CMU-Flownet: Exploring Point Cloud Scene Flow Estimation in Occluded Scenario

Jingze Chen
Center for Digital Media Computing
Xiamen University.
chenjingze@stu.xmu.edu.cn

Lei Li
Department of Computer Science
University of Copenhagen
lilei@di.ku.dk

Zerui Tang
Center for Digital Media Computing
Xiamen University.
bravetzr@stu.xmu.edu.cn

Qiqin Lin
Center for Digital Media Computing
Xiamen University.
qiqinl@stu.xmu.edu.cn

Junfeng Yao*
Center for Digital Media Computing
Xiamen University.
yao0010@xmu.edu.cn

Abstract

Occlusions hinder point cloud frame alignment in LiDAR data, a challenge inadequately addressed by scene flow models tested mainly on occlusion-free datasets.

Attempts to integrate occlusion handling within networks often suffer accuracy issues due to two main limitations: a) the inadequate use of occlusion information, often merging it with flow estimation without an effective integration strategy, and b) reliance on distance-weighted upsampling that falls short in correcting occlusion-related errors. To address these challenges, we introduce the Correlation Matrix Upsampling Flownet (CMU-Flownet), incorporating an occlusion estimation module within its cost volume layer, alongside an Occlusion-aware Cost Volume (OCV) mechanism. Specifically, we propose an enhanced upsampling approach that expands the sensory field of the sampling process which integrates a Correlation Matrix designed to evaluate point-level similarity. Meanwhile, our model robustly integrates occlusion data within the context of scene flow, deploying this information strategically during the refinement phase of the flow estimation. The efficacy of this approach is demonstrated through subsequent experimental validation. Empirical assessments reveal that CMU-Flownet establishes state-of-the-art performance within the realms of occluded Flyingthings3D and KITTY datasets, surpassing previous methodologies across a majority of evaluated metrics.

***Keywords:** Point Cloud, Scene Flow Estimaiton, Occluded Scenario*

1. Introduction

The advent of deep neural networks promotes scene flow estimation methodologies. A big breakthrough was realized with the inception of FlowNet3D [8], a paradigm that harnessed the foundational principles of PointNet++ [17] for the assimilation of local features into the fabric of scene flow estimation. This development marked the application of neural network architectures within this specialized domain. Subsequently, the development of HPLFlowNet [3] introduced an innovative mechanism for the computation of multi-scale correlations through the execution of upsampling operations embedded within bilateral convolutional layers. Building upon this, the work by [6] unveiled a pioneering technique aimed at learning a singular iteration of an unrolled iterative alignment procedure, thus enhancing the precision of scene flow estimations. The introduction of 3DFlow [24] heralded a new epoch in the domain, establishing new benchmarks in terms of 3D End Point Error (EPE3D) and overall accuracy metrics.

Despite the substantial advancements achieved by these

*Author of correspondence (Email: yao0010@xmu.edu.cn). The paper is supported by the National Natural Science Foundation of China (No. 62072388), Fujian Provincial Science and Technology Major Project (No. 2024HZ022003), Jiangxi Provincial Natural Science Foundation Key Project (No. 20244BAB28039), Xiamen Public Technology Service Platform (No. 3502Z20231043), and Fujian Sunshine Charity Public Welfare Foundation.

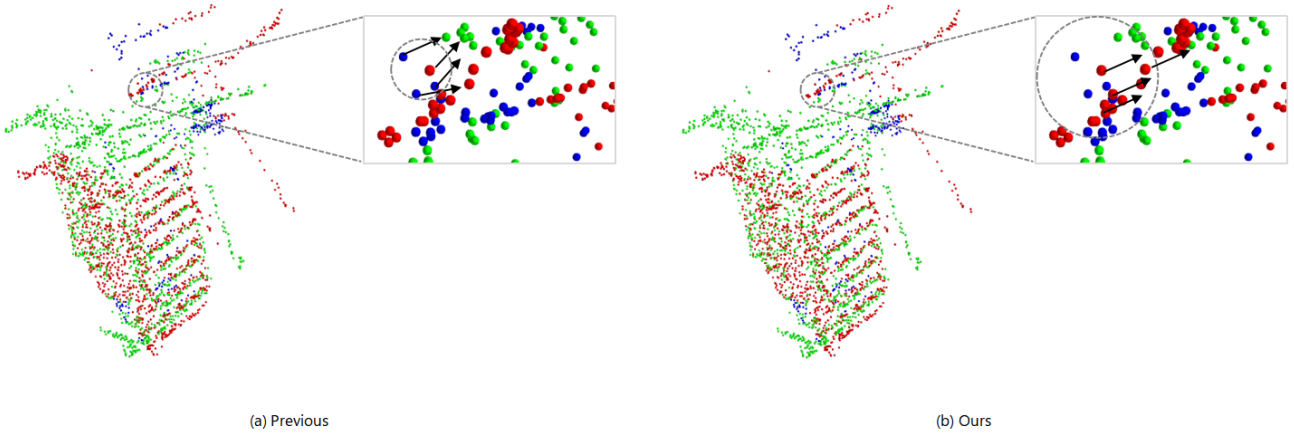

Figure 1. We illustrate the comparative methodology between conventional flow up-sampling techniques(left) and our proposed Matching Up-sampling framework(right), termed as CMU. Red points indicate positions at time t, green points represent positions at time $t+1$, and blue points signify areas that are occlued at time t. In contrast to previous methods that suffer from point mismatches due to their limited sampling scope, our CMU modules broaden the sampling range, and employs a correlation matrix to evaluate similarity at the point level to minimize errors.

neural network architectures, the challenge of occlusion remains a significant impediment. In the context of point-to-point matching tasks, occlusions introduce a critical issue: points visible at a given moment t may be obscured in the subsequent frame $t+1$ which may result in larger errors. In response to this pervasive challenge, Recent academic endeavors have concentrated on the integration of occlusion-aware mechanisms within the framework of neural network models, aiming to refine their understanding and processing of complex visual information by recognizing and accounting for occlusions. Zhao et al. [31] introduced an innovative asymmetric occlusion-aware feature matching module that is adept at learning a rudimentary occlusion mask. Such a mask is capable of filtering out regions rendered non-informative due to occlusion, immediately following feature warping processes. Further contributing to the discourse on occlusion mitigation, Saxena et al. [18] unveiled a pioneering self-supervised strategy aimed at the prediction of occlusions directly from image data. This methodology represents a paradigm shift towards leveraging inherent image characteristics to infer occlusion patterns, thus facilitating a more nuanced and accurate scene interpretation.

The integration of occlusion estimation with scene flow estimation in point clouds represents a significant stride towards addressing the occlusion challenges in dynamic 3D scenes. Ouyang et al. [14] pioneered this approach by advocating for the exclusion of computed Cost Volume for points identified as occluded, thereby mitigating the detrimental effects of occlusions on scene flow accuracy. This methodology marked the inception of occlusion-aware scene flow estimation, illustrating the feasibility and importance of occlusion consideration within this domain. Building upon this foundation, Wang et al. [26] further advanced the field by proposing a network architecture that features novel spatial and temporal abstraction layers, both augmented with an attention mechanism. This architecture also integrates an occlusion prediction module, enhancing the network's ability to discern and appropriately account for occluded regions within the point cloud. This addition not only improves the robustness of scene flow estimation against occlusions but also paves the way for more sophisticated handling of temporal and spatial dynamics. In a further evolution of this domain, Zhai et al. [30] introduced a cross-transformer model designed to capture more reliable dependencies between point pairs across frames. Integral to this model is the inclusion of occlusion predictions within both the network architecture and the loss function.

The integration of occlusion estimation with flow prediction methodologies has indeed marked a forward leap in tackling the complexities of dynamic 3D scene analysis. However, a discernible performance dichotomy persists between occluded and non-occluded dataset evaluations. Upon a meticulous examination of contemporary algorithms, several critical limitations have been identified, contributing to performance inefficacy within occluded environments. Firstly, there exists an overarching deficiency in the comprehensive exploitation of occlusion information. This is primarily manifested in the prevalent approach of coupling occlusion estimation with flow prediction tasks, which often lacks a nuanced strategy for their integration. Predominantly, recent models, Wang et al. [26] and Zhai et al. [30], employ a multitasking framework that concurrently executes occlusion and flow prediction, yet may not fully leverage the potential synergies between these tasks.

As noted in the studies by Cheng et al. [2] and Wang et al. [24], opt to overlook the information of occlusion, adopting a uniform treatment across all points within the scene. Moreover, while Ouyang et al. [14] innovatively apply occlusion data toward the refinement of Cost Volume feature extraction, the outcomes have yet to meet the anticipated benchmarks of efficacy.

Otherwise, the acquisition of multi-scale point cloud features is a cornerstone in the development of advanced scene flow estimation models, with a prevalent reliance on a coarse-to-fine paradigm for both downsampling and upsampling processes, as demonstrated in seminal works by Wu et al. [29], Ouyang et al. [14], and Zhao et al. [31]. This approach, while effective in a broad range of scenarios, predominantly utilizes a method of weighted upsampling based on Euclidean distances. Specifically, it involves the acquisition and weighted averaging of flow vectors from K neighboring points, a technique predicated on spatial proximities. However, this prevailing strategy exhibits drawbacks in the context of occluded datasets, where the simplistic nature of the upsampling mechanism can inadvertently amalgamate the flow of occluded points with those of unoccluded points. This scenario underscores a critical limitation in the current methodology, whereby the simplistic Euclidean-based upsampling fails to discern between occluded and non-occluded points, leading to an elevation in error rates within occluded scenarios.

Based on the shortcomings of previous models, as Figure.1 shown, we introduce the Correlation Matrix Upsampling Flownet (CMU-Flownet), a novel architecture which follows the coarse-to-fine paradigm. Our model incorporates an occlusion estimation module within its cost volume layer, alongside an Occlusion-aware Cost Volume (OCV) mechanism. Additionally, we propose an enhanced upsampling approach that expands the sensory field of the sampling process which integrate an Correlation Matrix designed to evaluate point-level similarity. Empirical evaluations demonstrate that CMU-Flownet sets a new benchmark for state-of-the-art performance in occluded Flyingthings3D and Kitti dataset. The key contributions of our study are outlined as follows:

- We introduce a new Occlusion-aware Cost Volume (OCV) methodology to detect the occluded points and perform feature extraction, passing the cost volume containing the occlusion information to the flow prediction module.

- We propose Correlation Matrix Upsampling (CMU) module based on geometric structures and point features. This is a plug-and-play module that can be integrated into any flow prediction task. Experiments show that our up-sampling structure is more accurate than the traditional approach.

- Our method outperforms previous pyramidal structures on the occluded Flyingthings3d and Kitti datasets, further improving the performance of the neural network in occluded scenarios.

2. Related Work

Scene Flow Estimation. The concept of scene flow was first articulated by Vedula et al. in [22]. Scene flow, distinct from the 2D optical flow that delineates the movement trajectories of image pixels, is conceptualized as a vector characterizing the motion of three-dimensional objects. Early research in this field [5, 11, 7, 23, 1] predominantly utilized RGB data. Notably, Huguet and Devernay [5] introduced a variational approach to estimate scene flow from stereo sequences, while Vogel et al. [23] presented a piece-wise rigid scene model for 3D flow estimation. Menze and Geiger [11] advanced the field by proposing an object-level scene flow estimation method, alongside introducing a dataset specifically for 3D scene flow. The advent of deep learning heralded transformative approaches in scene flow estimation. PointNet [16], as a pioneering work, utilized convolutional operations for point cloud feature learning, which was further refined by PointNet++ [17] through feature extraction from local domains. Subsequent studies [8, 15, 29, 24] have achieved impressive results in scene flow estimation. FlowNet3D [8], for instance, leverages PointNet++ [17] for feature extraction and introduces a flow embedding layer to capture and propagate correlations between point clouds for flow estimation. Puy et al. [15] employed optimal transport for constructing point matches between sequences. Wu et al. [29] proposed a cost volume module for processing large motions in 3D point clouds, while Wang et al. [24] innovated an all-to-all flow embedding layer with backward reliability validation to address consistency issues in initial scene flow estimation.

Cost Volume. The advent of Cost Volume as a transformative tool in the optical flow domain has catalyzed innovative developments in scene flow estimation. Wu et al. [29] were at the forefront of this evolution, pioneering the integration of a Cost Volume module that predicts scene flow by constructing cost volumes at each level of the feature pyramid. To effectively accommodate large motions, the PointPWC-Net introduced a coarse-to-fine strategy, which involves concatenating features at level L with the upsampled features from level $L+1$, thereby enhancing motion capture capabilities across different scales. Building upon this foundational pyramid structure, subsequent research endeavors have sought to refine and extend the utility of the cost volume concept. Wei et al. [28] proposed a groundbreaking approach that utilizes correlation volumes as a means to circumvent the limitations inherent in previous cost-volume based methodologies, specifically targeting the mitigation of error accumulation issues. Further,

Cheng et al. [2] drew inspiration from the upsampling and warping layers of PointPWC-Net [29], applying these techniques to enhance the fidelity of scene flow predictions.In a significant leap forward, Wang et al. [24] introduced an innovative all-to-all flow embedding layer, accompanied by a backward reliability validation mechanism. This approach is designed to tackle consistency challenges encountered in initial scene flow estimations, thereby setting new benchmarks for performance within the field at the time of its introduction.

Occlusion in Flow Estimation. The domain of optical flow estimation has witnessed significant advancements through the adept handling of occlusions. Drawing inspiration from these successes, Ouyang et al. [14] embarked on an innovative endeavor to harness occlusion data for enhancing Cost Volume feature extraction methodologies. Despite these efforts, the achieved outcomes have yet to fulfill the anticipated efficacy benchmarks. This has led to the incorporation of occlusion prediction modules within network architectures emerging as a pivotal strategy for augmenting accuracy in flow estimation tasks.Subsequent developments have seen scholars like Wang et al. [26] and Zhai et al. [30] integrating this methodology into their models, thereby embedding occlusion prediction as a component of the loss function. Empirical evaluations of this approach have validated its effectiveness. This dual-faceted impact, wherein occlusion information for each point within the point cloud is ascertainable, coupled with the synergistic benefits of a multi-task fusion network architecture, fosters a conducive environment for the point convolutional layers. [27] adopts a subnet to predict the occlusion mask and explicitly masks those occluded points, which ensures flow predictor to focus on estimating the motion flows of non-occluded points. [9] propose a module based on the transformer, which utilizes local and global semantic similarity to infer the motion information of occluded points.

3. Problem Formulation

Given two sequential point clouds of the identical scene, represented as $P = \{(x_i, p_i) \in \mathbb{R}^3 | i = 1, 2, \ldots, n\}$ and $Q = \{(y_j, q_j) \in \mathbb{R}^3 | j = 1, 2, \ldots, n\}$, where x_i and y_j are the coordinates of points in P and Q respectively, and p_i and q_j represent the feature attributes (such as color, normal vectors) at two different time frames. The objective is to compute a 3D motion field $F = \{f_i \in \mathbb{R}^3 | i = 1, 2, \ldots, n\}$, which specifies the transformation vectors needed to align P onto Q. This involves determining an optimal permutation matrix M from the set $\{0, 1\}^{n \times n}$ to satisfy the equation $P + F = MQ$, aiming to closely approximate the motion field F to the ground truth F_{gt} with high accuracy.

Concurrently, the analysis endeavors to ascertain the occlusion status $O(x_i)$ for each point x_i originating from the first frame point cloud. Here, $O(x_i) = 1$ indicates that the point x_i is unoccluded, while $O(x_i) = 0$ indicates occlusion. Distinguishing between occluded and non-occluded states is crucial for improving the accuracy of the 3D motion field prediction, which allows for the adaptation to dynamic occlusion scenarios that are common in sequential point cloud data.

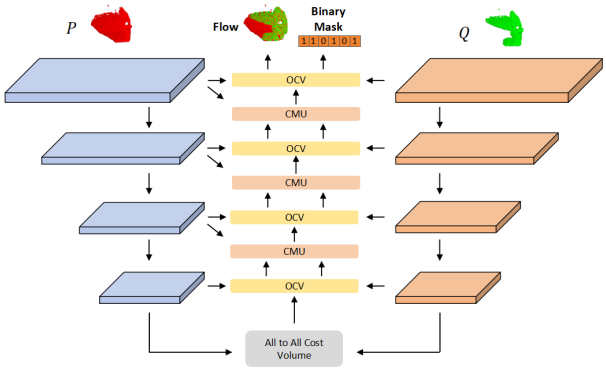

Figure 2. We take the traditional pyramid-type structure as the overall framework of our model. Each frame of point cloud data is processed through a hierarchical point feature abstraction module consisting of four layers. Then a Correlation Matrix Upsampling (CMU) module is employed for upsampling purposes, and Occlusion-aware Cost Volume (OCV) is used for further refining the flow. After iterating through these processes for a specified number of loops, the model outputs the final predicted 3D motion field.

4. Method

We take the traditional pyramid-type structure as the overall framework of our model which is proven to be efficient in flow estimation task [29, 14, 24]. We take Q and P as inputs. Each frame of the point cloud data is processed through a hierarchical point feature abstraction module that consists of four layers. The abstraction process at each layer employs Farthest Point Sampling (FPS) for downsampling and uses PointConv [17]. The spatial coordinates of the points at layer l are denoted by x^l and y^l. Feature inheritance is performed from the previous layer $l-1$, resulting in the derived features p^l and q^l. This derivation involves operations such as grouping, pooling, and the application of weight-shared Multilayer Perceptrons (MLP). The sampling ratio at each layer is set to be 1/4 of the preceding layer, effectively reducing the number of points processed and refined at each subsequent stage. The overall architecture of CMU-Flownet is shown in 2.

After extracting features from successive Pointconv layers, we adopt a coarse-to-fine paradigm to obtain scene flow at different scales progressively from one layer to the next. Motivated by [24], we first apply the All-to-All Cost Volume at the bottom level of our network to build a correlation

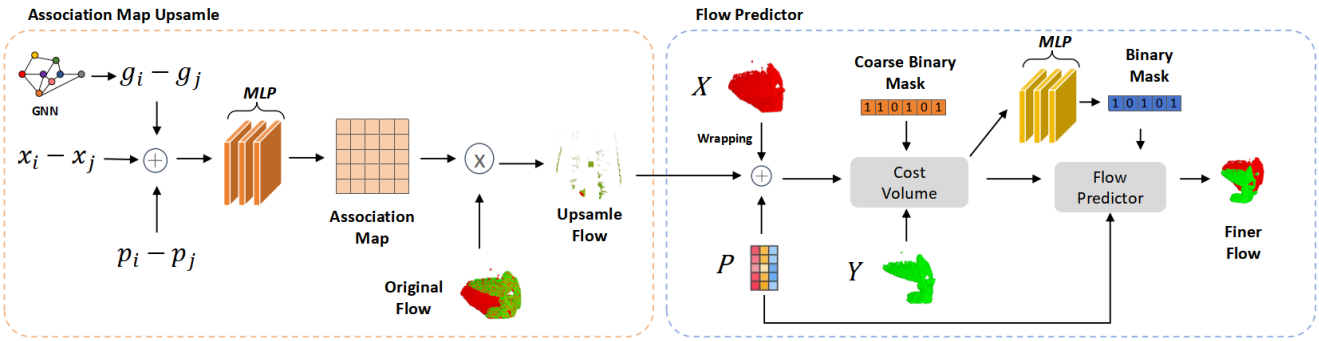

Figure 3. The picture shows the Correlation Matrix Upsampling (CMU) on the left and Flow Predictor on the right. Occlusion-aware Cost Volume (OCV) is the feature extractor in Flow Predictor. We use CMU for upsample flow and refine the flow by Flow Predictor.

between two frames and learn the flow embedding. We then used the Correlation Matrix Upsampling (CMU) module for upsampling. We then add our proposed Occlusion-aware Cost Volume (OCV) to further refine the flow. Compared to the previous Cost Volume module, we add a masking prediction component and incorporate the masking information into the attention mechanism to obtain a more efficient cost volume. After a few loops output the final result, the predicted scene flow F and the Binary mask $O(x_i)$.

In this section, we mainly discuss the proposed CMU and OCV modules. Detailed implementation information and schematic diagrams for all components discussed are provided in the supplementary materials.

4.1. Correlation Matrix Upsampling

Most point cloud processing methods follow a coarse-to-fine paradigm. Various upsampling strategies have been used to construct scene flow fields from sparse levels to dense levels with proper weights. some work like [8, 29, 14] et. al. predict flow based on trilinear interpolation upsampling, it is simple and efficient, but can lead to error accumulation, especially in occluded datasets. [4, 25] proposed an intra-frame patch features based method which uses interpolation functions to represent the distances between each point and its neighboring points. The proposal of this method has improved some accuracy, but it is still unable to adapt to complex occlusion Scenario and larger sensory field. Inspired by [19] which use features to update superpoints, we productively propose an enhanced upsampling approach that expands the sensory field of the sampling process which integrate an Correlation Matrix designed to evaluate point-level similarity.

Our method attempts to generate an upsampling flow that satisfies the following requirements: (1) Neighbouring points are with similar flow patterns; (2) Points with similar characteristics share similar flow. Thus, we introduce graph-structure based coding on the point cloud to extract point cloud features. First, we construct the graph structure at the point level, where the edges of the graph aggregate information such as the position of the points, colour and normal vector. Then we use setconv layer as encoder to learn neighbourhood information, Combine the previously obtained features p^l and q^l. Subsequently, we use MLP to learn the similarity of different point features.

4.1.1 Flow-Graph Encoder

Graph Neural Networks (GNNs) are a category of neural networks designed specifically for processing data structured as graphs. Graphs are mathematical structures used to model pairwise relations between objects, characterized by vertices (nodes) and edges (links). Work in [20] demonstrates the effectiveness of GNN in point cloud processing tasks.

We take the upsampling process from $l+1$ to l layer as an example to illustrate the Correlation Matrix calculation. Next, we define the graph structure as follows:

$$E^l = \{(x_i^l - x_j^{l+1} || p_i^l || p_j^{l+1}) \mid \|x_i^l - x_j^{l+1}\|_2 < r\} \quad (1)$$

Equation 1 represents the establishment of the graph structure. In Equation 1, E denotes the edge set in the graph structure. We select the neighboring points around each point and use their distance and feature differences as the criteria for edge formation. p_i^{l+1} and p_j^l denote the feature in two different layers. $||$ is the contact operator. Following the methodology suggested by Kittenplon et al. [6], we encode edge features using three consecutive $setconv$ layers as our convolution mechanism.

In contrast to previous GNN methods, we introduce the concept of spatial memory in the feature extractor. Spatial memory has been effectively applied in the field of semantic segmentation, where studies such as [21, 13] demonstrate that sequential input outperforms single-frame input by enabling the neural network to incorporate temporal information. While methods by [29, 14] utilize wrapped points to gather neighborhood information across different frames, our model diverges by leveraging this temporal and spatial information specifically for flow upsampling. In the context

of scene flow estimation, which inherently carries temporal data, our model attaches the coarse flow to point P at time $t-1$ and retains memory up to point Q at time t, thus facilitating the learning of geometric information across time intervals.

$$
\begin{aligned}
x^l_{w,i} &= x^l_i + f^l_{c,i} \\
E^l_w &= \{(x^l_{w,i} - y^{l+1}_j || p^l_i || q^{l+1}_j) \mid \|x^l_{w,i} - y^{l+1}_j\|_2 < r\}
\end{aligned} \tag{2}
$$

At the beginning, we take the traditional approach to get the initial coarse flow $f^{l-1}_{c,i}$. And $x_{w,i}$ denotes the i^{th} point in P wrapped by coarse flow, E^l_w indicates the features of point $p_{w,i}$ with the memory module, and we keep points to the next frame to learn the features of the surrounding points. We use $setconv$ layers to encode the features. g and $g_{w,i}$ denotes the finished encoded feature. As previously articulated in the formula, $g^l_{w,i}$ denotes the feature which is obtained from wrapped points.

$$
\begin{aligned}
g^l_i &= setconv(e^l_i),\ e_i \in E^l \\
g^l_{w,i} &= setconv(e^l_{w,i}),\ e^l_{w,i} \in E^l_w
\end{aligned} \tag{3}
$$

4.1.2 Correlation Matrix

Flow consistency algorithms based on distance only can lead to prediction errors because the motion patterns of object boundary points are very different from those of surrounding points. So we model based on Euclidean distances and feature distances, the purpose of which is to reduce errors. Let I denotes the total number of points in layer l, and N denotes the N nearest neighbouring points around x^l_i (We set $N = 32$). Correlation Matrix is a module for learning point cloud similarity based on a feature encoder, where we combine the temporal and spatial features learned in the previous section to generate an $I \times N$ matrix that measures the similarity between point levels.

$$
\begin{aligned}
u_{i,n} &= (x^l_i || x^l_{w,i}) - (x^{l+1}_n || x^{l+1}_{w,n}) \\
v_{i,n} &= (g^l_i || g^l_{w,i}) - (g^{l+1}_n || g^{l+1}_{w,n}) \\
w_{i,n} &= (p^l_i || p^l_{w,i}) - (p^{l+1}_n || p^{l+1}_{w,n}) \\
a_{i,n} &= MLP(u_{i,n}) + MLP(v_{i,n}) + MLP(w_{i,n})
\end{aligned} \tag{4}
$$

Where $a_{i,n}$ denotes the degree of similarity between the i^{th} layer l point and the n^{th} layer $l+1$ point. Next, we assign each point p_i a similarity vector, We map the similarity parameter to the interval $[0, 1]$ as a weight for flow upsampling.

$$
a_{i,n} = softmax([a_{i,1}, a_{i,2}..., a_{i,N}])_n \tag{5}
$$

We update the scene flow vector with the Correlation Matrix that maps the ground truth labels to each point, $f^l_{u,i}$ means the i^{th} upsampling flow.

$$
f^l_{u,i} = \sum_{n=1}^{N} a_{i,n} * f^{l+1}_n \tag{6}
$$

The inclusion of correlation and weighted summation ensures that the upsampling process is both context-aware and spatially precise, thereby improving the fidelity of scene flow estimations in applications like 3D scene reconstruction and motion analysis.

4.2. Occlusion-aware Cost Volume

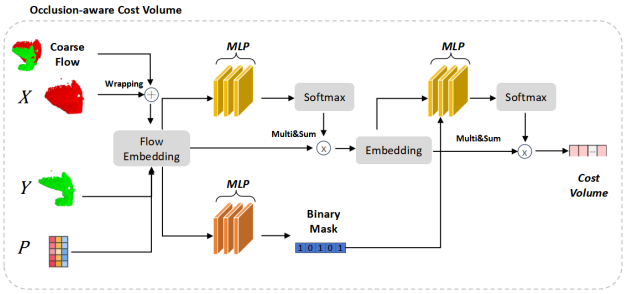

Figure 4. Overview of Occlusion-aware Cost Volume (OCV) module.

Cost Volume, in Figure. 4, is a measure of correlation between two frames of the point cloud. Previous work has confirmed its validity. We introduce an attention mechanism that includes occlusion information based on past models. Taking the computation of cost volume in layer l as an example, we proceed to describe its computation steps.

The inputs are the coordinates and features of point cloud P and Q in layer l. Otherwise, we added the occlusion $O(x_i)$ as input to capture information in different dimensions. We first calculate the occlusion formula.

$$
\begin{aligned}
cost(x^l_i, y^l_j) &= MLP(x^l_i - y^l_j || p^l_i || q^l_j) \\
O^l(x_i) &= Sigmoid(\underset{||x^l_i - y^l_j|| < r}{MAX} \{cost(x^l_i, y^l_j)\})
\end{aligned} \tag{7}
$$

Next, we learn the point cloud matching information for two frames. Inspired by [25], we compute the two-frame point cloud cost volume and then perform weighted average pooling for this data in P^l, which is to reduce the error due to long distance matching.

$$
\begin{aligned}
CV_1(x^l_i) &= \sum_{||x^l_i - y^l_j|| < r} Weight_1(x^l_i, y^l_j) * MLP(cost(x^l_i, y^l_j)) \\
CV(x^l_i) &= \sum_{||x^l_i - x^l_j|| < r} Weight_2(x^l_i, x^l_j, O^l_i) * MLP(CV_1(x^l_j))
\end{aligned} \tag{8}
$$

Where $Weight_1$ and $Weight_2$ denote the weights function of each of the N proximity points about point xi. As

before, || denotes matrix contact operation. For simplicity, O_i^l represents $O(x_i)$.

4.3. Flow Refinement

We compute the obtained cost volume value, which is used to refine the previously obtained upsampled flow $f_{u,i}^l$. In addition to the cost volume, we add the upsampled flow $f_{u,i}^l$, the occlusions $O(x_i)$, and the feature information p_i^l. Unlike [25, 24], we discard using flow encoding as the information and add the occlusion information, and experiments show that our module reduces the computational cost while improving the accuracy.

$$\begin{aligned} \Delta f_i^l =& MLP(CV(x_i^l)||p_i^l||f_{u,i}^l||O(x_i)) \\ f_i^l =& f_{u,i}^l + \Delta f_i^l \end{aligned} \tag{9}$$

4.4. Loss Fuction

At each layer, we can obtain the estimated occlusion $O(x_i^l)$ and flow f_i^l. We adopt a cyclic strategy to compute the loss function for each layer and attach appropriate weights. We divide the loss into two components, the occlusion loss as well as the flow loss which is similar to [14].

$$\begin{aligned} Loss_o &= \sum_{l=0}^{3} \beta^l * \|O(x_i^l) - O_{gt}(x_i^l)\|_2 \\ Loss_f &= \sum_{l=0}^{3} \beta^l * \|f_i^l - f_{gt,i}^l\|_2 \\ Loss &= \alpha * Loss_f + (1-\alpha) * Loss_o \end{aligned} \tag{10}$$

Where $Loss_o$ and $Loss_f$ represent the occlusion loss and flow loss respectively.

5. Experiment

5.1. Experimental Setups

Dataset We conduct our experiments on $FT3D_o$[10] and $KITTI_o$[11, 12] respectively. FlyingThings3D [10] is a synthetic dataset for optical flow, disparity and scene flow estimation. It consists of everyday objects flying along randomized 3D trajectories. In the field of point cloud scene flow estimation, there are two commonly used data processing methods. The first version is prepared by HPLFlowNet [4], we denote these datasets without occluded points as $FT3D_s$. The second version is prepared by Flownet3D [8], we denote this occluded dataset as $FT3D_o$. KITTI is a real-world scene flow dataset with 200 pairs for which 142 are used for testing without any fine-tuning. The KITTI can also be divided into occluded and non-occluded versions, named $KITTI_s$ and $KITTI_o$, respectively. To verify the effectiveness of our model in occluded scenes, we take the processing in Flownet3D [8] to generate the occluded datasets.

Details Our model is trained based on pytorch, using NVIDIA GeForce RTX 3090 as the hardware device. we train our model on synthetic $FT3D_o$ training data and evaluate it on both $FT3D_o$ test set and $KITTI_o$ without finetune. Referring to most practices in the domain, we randomly take 8192 points per batch in training. In terms of model parameters, we set the upsampling range $N = 32$ and the hyperparameters β^l as [0.02, 0.04, 0.08, 0.16], α as 0.8 for training. We set the learning rate to 0.001 and the decay factor to 0.5, and decay in every 80 training epochs. We take Adam as the optimizer with default values for all parameters. In total, we train about 400 epochs.

Evaluation Metrics We test our model with four evaluation metrics, including End Point Error (EPE), Accuracy Strict (AS), Accuracy Relax (AR), and Outliers (Out). We denote the estimated scene flow and ground truth scene flow as F and F_{gt}, respectively. EPE(m): $\|F - F_{gt}\|_2$ averaged over all points. AS: the percentage of points whose EPE <0.05m or relative error<5%. AR: the percentage of points whose EPE <0.1m or relative error <10%. Out: the percentage of points whose EPE > 0.3m or relative error > 10%.

Table 1. Comparison of our model with previous methods on the occluded datasets $FT3D_o$. In these models, DELFlow uses a multimodal training approach where they use point clouds and images as input.

Method	sup.	EPE↓	AS↑	AR↑	Out↓
FlowNet3D	full	0.169	0.254	0.579	0.789
FLOT	full	0.156	0.343	0.643	0.700
PointPWC-Net	full	0.155	0.416	0.699	0.639
OGSFNet	full	0.122	0.552	0.777	0.518
FESTA	full	0.111	0.431	0.744	-
FlowFormer	full	0.077	0.720	0.866	0.316
CamLiRAFT	full	0.076	0.794	0.904	0.279
3DFlowNet	full	0.063	0.791	0.909	0.279
Ours	full	**0.052**	**0.843**	**0.927**	**0.212**

Table 2. Comparison of our proposed method with previous methods on the occluded datasets $KITTI_o$. As previous method do, we train on $FT3D_o$ and test on $KITTI_o$ without any finetune.

Method	sup.	EPE↓	AS↑	AR↑	Out↓
FlowNet3D	full	0.173	0.276	0.609	0.649
FLOT	full	0.110	0.419	0.721	0.486
PointPWC-Net	full	0.118	0.403	0.757	0.497
OGSFNet	full	0.075	0.706	0.869	0.328
FESTA	full	0.094	0.449	0.834	-
FlowFormer	full	0.074	0.784	0.883	0.262
3DFlowNet	full	0.073	0.819	0.890	0.261
Ours	full	**0.065**	**0.856**	**0.911**	**0.221**

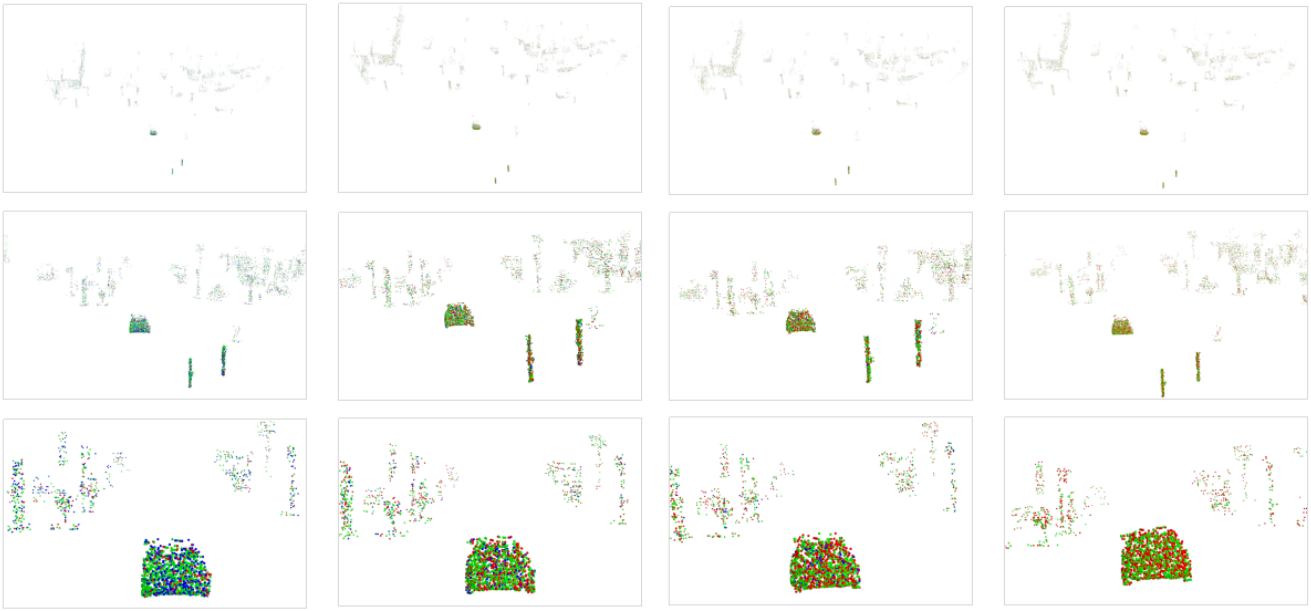

Figure 5. Visualization on KITTI$_o$, red points represent the position at time t wrapped by predicted flow, green represent the position at time $t+1$, and blue points indicate inaccurate predictions (measured by Acc3D Strict).

5.2. Performance On Occluded Dataset

We train our model in FT3D$_o$, at the same time we tested on the FT3D$_o$ test set and KITTI$_o$, the results are shown in Table 1. We compare our performance with mainstream models in recent years, and our approach outperforms all other methods (Note that we compare under the occluded dataset). Our network improve performance by about 57.4% compared to OGSFNet[14], a previous model for occluded environments. Otherwise, we surpass 3DFlownet[24] by about 21.2%, which is the optimal algorithm in recent years. Moreover, in a comparison with the multimodal method DELFlow , we were able to exceed its performance when using only the point cloud as input. Also, the experimental results on KITTI prove that our model has generalisation ability. On EPE3D, we outperform 3DFlownet[24] 0.008. And there is a considerable improvement in AS and AR, compared to the previous advanced method, we improve performance by about 4.5% and 2.4%. On Outliers , we reduce the error rate by about 3% compared with the previous one, and such results show that our model achieves a leading level in error control.

For the occlusion estimation, our model achieves a 93.6% accuracy on the FT3D$_o$, it shows that our model can effectively perceive the occlusion and avoid the error generated by the occlusion, which is one of the reasons for the efficient performance of our model.

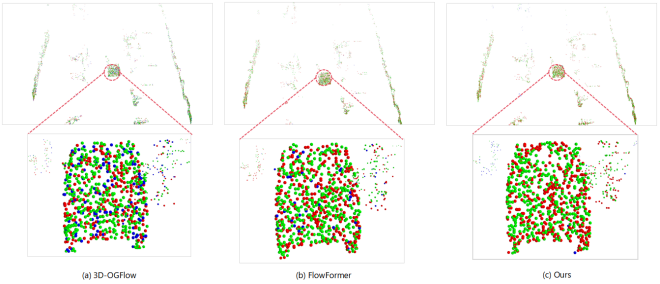

Figure 6. Visualization on KITTI$_o$, red points represent the position at time t wrapped by predicted flow, green represent the position at time t + 1, and blue points indicate inaccurate predictions (measured by Acc3D Strict).

5.3. Performance in Non-occluded Dataset

We train our model in quarter of FT3D$_s$, the results are shown in Table 3 and 4. We compare our performance with mainstream models in recent years, and our approach yielded good results.

We keep the original model unchanged and remove the part of the loss function about the occlusion information. The results of our model are shown in 3 and 4. We can see that we can still achieve good results in the non-occluded dataset.

5.4. Ablation Study

To validate the effectiveness of our key components, we perform ablation experiments on two proposed modules.

Table 3. Comparison of our model with previous methods on the non-occluded datasets $FT3D_s$.

Method	sup.	EPE↓	AS↑	AR↑	Out↓
FlyingThings3D					
FlowNet3D	full	0.177	0.374	0.668	0.527
FLOT	full	0.056	0.755	0.908	0.242
PointPWC-Net	full	0.069	0.728	0.888	0.265
FlowStep3D	full	0.055	0.805	0.925	0.149
HALFlow	full	0.062	0.765	0.903	0.249
OGSFNet	full	0.036	0.879	-	0.197
Ours	full	**0.031**	**0.913**	**0.977**	**0.158**

Table 4. Comparison of our model with previous methods on the non-occluded datasets $FT3D_s$.

Method	sup.	EPE↓	AS↑	AR↑	Out↓
Kitti					
FlowNet3D	full	0.114	0.413	0.771	0.602
FLOT	full	0.052	0.732	0.927	0.357
PointPWC-Net	full	0.059	0.738	0.928	0.342
FlowStep3D	full	0.046	0.816	0.961	0.217
HALFlow	full	0.051	0.781	0.944	0.309
OGSFNet	full	0.038	0.882	-	0.175
Ours	full	**0.034**	**0.893**	**0.944**	**0.165**

First we replace the CMU(Correlation Matrix Upsampling) based upsampling algorithm with the most common trilinear upsampling. Secondly, we use the Cost Volume module, which also integrates occlusion prediction, to compare with our OCV(Occlusion-aware Cost Volume). We replace OCV in the model with Cost Volume in [14] and test it. We sequentially demonstrate the effectiveness of CMU and OCV by arranging and combining the modules of the existing model with the traditional approach of the past. We train the replaced model and the results are shown in table 5.

Table 5. The ablation experiment focuses on the two proposed modules. We replace the CMU module with trilinear interpolation upsampling and the OCV using the Cost Volume calculation method in [14]. The experimental results show the effectiveness of our modules.

Data.	CMU	OCV	EPE↓	AS↑	AR↑	Out↓
Fly.			0.065	0.777	0.905	0.301
		✓	0.061	0.790	0.914	0.290
	✓		0.057	0.816	0.920	0.240
	✓	✓	0.052	0.843	0.927	0.212
KI.			0.807	0.810	0.881	0.263
		✓	0.077	0.818	0.883	0.252
	✓		0.072	0.849	0.904	0.226
	✓	✓	0.065	0.856	0.911	0.221

As shown in Table 5, when we discarded two modules, the metrics declined to varying degrees. First, when we do not use CMU and OCV, EPE3D rises to 0.065, and the accuracy of each is reduced by 2-4%. When we introduced OCV, EPE3D was elevated by about 6%, and accuracy AS and AR also improved somewhat. When we added CMU, the model performance was improved compared to the past, with EPE3D, AS,AR, improved by 12.3%, 5%, and 1.5%, respectively. The best performance can be obtained when we add two modules at the same time. The same result can be seen on KITTI dataset. Although the degree of decrease in epe3d is not significant, we have a substantial improvement in both AS and AR due to the effect of the finer upsampling, which shows that our module is still valid in the real-world dataset, as well.

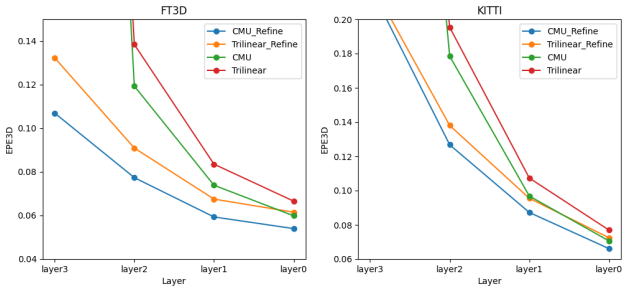

Figure 7. The EPE values of the predicted flow vectors for each layer are shown in Fig. The two lines represent the epe values of the flows obtained by CMU(Correlation Matrix Upsampling) and trilinear interpolation upsampling, respectively. From the figure, it can be seen that the up-sampled flow obtained through CMU has higher accuracy.

To further illustrate the role of the CMU in upsampling, we measure the error of the flow computed by each layer. Again we use a traditional linear method and CMU to compare the accuracy of both methods before and after flow refinement. The graphs illustrate that our module improves the accuracy of the streams at all layers and can be integrated into any model from coarse to fine.

5.5. Sampling Range Setting

Table 6. The graphs illustrate the accuracy of the model at different sampling ranges in FT3D dataset.

N	EPE↓	AS↑	AR↑	Out↓
8	0.059	0.813	0.918	0.249
16	0.056	0.819	0.920	0.236
32	0.052	0.843	0.927	0.212
64	0.059	0.794	0.912	0.266

In this subsection we explore the effect of the value of the number of samples N in cmu on the final results. If we use a smaller range, it may lead to errors due to occlusion, and if the range is larger it will lead to a decrease in computational power and accuracy. We train on flying to explore the change in EPE3D for each layer and the final results when only the sampling range is changed.

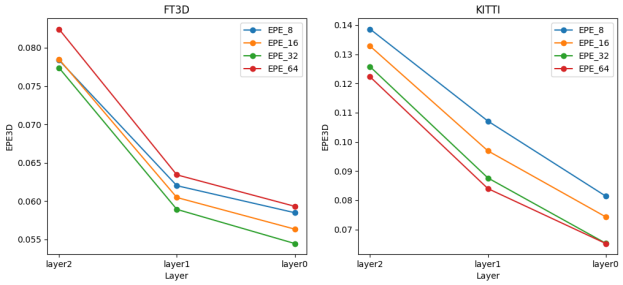

Figure 8. The line graph illustrates the EPE3D values for each layer of the network structure when $N = 8, 16, 32, 64$.

From the graphs and tables, the effect on the accuracy of the model at different N settings can be obtained. We conducted experiments in FT3D and KITTI respectively. On the FT3D dataset, the best accuracy is possessed when set to $N = 32$, and a large drop in accuracy occurs when $N = 8$ and 64 due to too small and too large sensory fields. On the KITTI dataset, $N = 64$ has the best accuracy, this is because in real scene datasets the points have a large distance to move and when we expand the receptive field it gives better results. Combining the performance of the above two datasets, we take $N = 32$, partly because we can get better model accuracy and partly because the time loss is smaller.

5.6. Weights in Correlation Matrix

We analysed the weights of the Correlation Matrix to illustrate the degree of preference for different points in our CMU module.

Table 7. Evaluation in average weights of non-occluded points and occluded points.

Metrics	Non-occluded Points	Occluded Points
Weight	0.003245	0.000661

We separately explore the weights of occluded and non-occluded points in the up-sampling process, and these weights illustrate the extent to which each point influences the flow of the scene in high resolution. As can be seen from the table 7, the weights of the non-occluded points are higher than the weights of the occluded points, which reduces the error due to occlusion.

5.7. Assessment of Model Efficiency

The efficiency of a model has been seen as an important part of evaluating model performance in recent years. Due to the limitation of equipment performance, how to make the model lightweight has also received much attention. We compare the model size and run time with the current mainstream models to show that the efficiency of our model is within a reasonable range. All models were experimented on the same hardware.

Table 8. Evaluation of model size and run time.

Metrics	FlowNet3D	HPLFlowNet	FESTA	3DFlownet	**ours**
Size (MB)	14.9	231.8	16.1	19	22
Time (ms)	34.9	93.1	67.8	60.1	64.5

We select [8, 4, 26, 24] as a comparison to our model. From table 8 we can see that the efficiency of our model is better than HPLFlowNet [4] and slightly lower than the remaining ones. However, our accuracy is large higher than these methods, and the appropriate sacrifice of efficiency is acceptable.

6. Conclusion

In this study, we address the challenge of robustly matching successive frames in point cloud sequences. Despite the recognized potential of neural network-based scene flow estimation, its application in occlusion-rich environments remains partially explored. To advance this area, we introduce the Correlation Matrix based Upsampling Flownet (CMU-Flownet), that seamlessly integrates an occlusion estimation module within its cost volume layer via an Occlusion-aware Cost Volume (OCV) mechanism. Additionally, our model incorporates a novel upsampling strategy utilizing a Correlation Matrix to evaluate point-level similarity. Through rigorous empirical evaluations on datasets known for their occluded scenarios, such as Flyingthings3D and Kitti, CMU-Flownet demonstrates superior performance over existing methods across various metrics.

References

[1] J. Chen, J. Yao, Q. Lin, R. Zhou, and L. Li. Ssflownet: Semi-supervised scene flow estimation on point clouds with pseudo label. *arXiv preprint arXiv:2312.15271*, 2023. 3

[2] W. Cheng and J. H. Ko. Bi-pointflownet: Bidirectional learning for point cloud based scene flow estimation. In *European Conference on Computer Vision*, pages 108–124. Springer, 2022. 3, 4

[3] X. Gu, Y. Wang, C. Wu, Y. Lee, and P. Wang. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. *Cornell University - arXiv,Cornell University - arXiv*, Jun 2019. 1

[4] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3254–3263, 2019. 5, 7, 10

[5] F. Huguet and F. Devernay. A variational method for scene flow estimation from stereo sequences. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–7. IEEE, 2007. 3

[6] Y. Kittenplon, Y. Eldar, and D. Raviv. Flowstep3d: Model unrolling for self-supervised scene flow estimation. *Cornell University - arXiv,Cornell University - arXiv*, Nov 2020. 1, 5

[7] L. Li. Hierarchical edge aware learning for 3d point cloud. In *Computer Graphics International Conference*, pages 81–92. Springer, 2023. 3

[8] X. Liu, C. R. Qi, and L. J. Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 529–537, 2019. 1, 3, 5, 7, 10

[9] Z. Lu and M. Cheng. Gma3d: Local-global attention learning to estimate occluded motions of scene flow. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 16–27. Springer, 2023. 4

[10] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. 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. 7

[11] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3061–3070, 2015. 3, 7

[12] M. Menze, C. Heipke, and A. Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140:60–76, 2018. 7

[13] D. Nilsson and C. Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6819–6828, 2018. 5

[14] B. Ouyang and D. 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, 3, 4, 5, 7, 8, 9

[15] G. Puy, A. Boulch, and R. Marlet. Flot: Scene flow on point clouds guided by optimal transport. *Cornell University - arXiv,Cornell University - arXiv*, Jul 2020. 3

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

[17] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1, 3, 4

[18] R. Saxena, R. Schuster, O. Wasenmuller, and D. Stricker. Pwoc-3d: Deep occlusion-aware end-to-end scene flow estimation. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 324–331. IEEE, 2019. 2

[19] Y. Shen, L. Hui, J. Xie, and J. Yang. Self-supervised 3d scene flow estimation guided by superpoints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5271–5280, 2023. 5

[20] W. Shi and R. Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1711–1719, 2020. 5

[21] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4481–4490, 2017. 5

[22] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 722–729. IEEE, 1999. 3

[23] C. Vogel, K. Schindler, and S. Roth. 3d scene flow estimation with a rigid motion prior. In *2011 International Conference on Computer Vision*, pages 1291–1298. IEEE, 2011. 3

[24] G. Wang, Y. Hu, Z. Liu, Y. Zhou, M. Tomizuka, W. Zhan, and H. Wang. What matters for 3d scene flow network. In *European Conference on Computer Vision*, pages 38–55. Springer, 2022. 1, 3, 4, 7, 8, 10

[25] G. Wang, X. Wu, Z. Liu, and H. Wang. Hierarchical attention learning of scene flow in 3d point clouds. *IEEE Transactions on Image Processing*, 30:5168–5181, 2021. 5, 6, 7

[26] H. Wang, J. Pang, M. A. Lodhi, Y. Tian, and D. Tian. Festa: Flow estimation via spatial-temporal attention for scene point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14173–14182, 2021. 2, 4, 10

[27] K. Wang and S. Shen. Estimation and propagation: Scene flow prediction on occluded point clouds. *IEEE Robotics and Automation Letters*, 7(4):12201–12208, 2022. 4

[28] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6954–6963, 2021. 3

[29] W. Wu, Z. Wang, Z. Li, W. Liu, and L. Fuxin. Pointpwc-net: A coarse-to-fine network for supervised and self-supervised scene flow estimation on 3d point clouds. *Cornell University - arXiv,Cornell University - arXiv*, Nov 2019. 3, 4, 5

[30] M. Zhai, K. Ni, J. Xie, and H. Gao. Learning scene flow from 3d point clouds with cross-transformer and global motion cues. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 2, 4

[31] S. Zhao, Y. Sheng, Y. Dong, E. I. Chang, Y. Xu, et al. Maskflownet: Asymmetric feature matching with learnable occlusion mask. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6278–6287, 2020. 2, 3