RMS-FlowNet: Efficient and Robust Multi-Scale Scene Flow Estimation for Large-Scale Point Clouds

Ramy Battrawy[1], René Schuster[1], Mohammad-Ali Nikouei Mahani[2] and Didier Stricker[1]

***Abstract*— The proposed RMS-FlowNet is a novel end-to-end learning-based architecture for accurate and efficient scene flow estimation which can operate on point clouds of high density. For hierarchical scene flow estimation, the existing methods depend on either expensive Farthest-Point-Sampling (FPS) or structure-based scaling which decrease their ability to handle a large number of points. Unlike these methods, we base our fully supervised architecture on Random-Sampling (RS) for multi-scale scene flow prediction. To this end, we propose a novel flow embedding design which can predict more robust scene flow in conjunction with RS. Exhibiting high accuracy, our RMS-FlowNet provides a faster prediction than state-of-the-art methods and works efficiently on consecutive dense point clouds of more than 250K points at once. Our comprehensive experiments verify the accuracy of RMS-FlowNet on the established FlyingThings3D data set with different point cloud densities and validate our design choices. Additionally, we show that our model presents a competitive ability to generalize towards the real-world scenes of KITTI data set without fine-tuning.**

I. Introduction

Scene flow estimation is a key computer vision task for the purposes of navigation, planning tasks and autonomous driving systems. It concerns itself with the estimation of a 3D motion field with respect to the observer, thereby providing a representation of the dynamic change in the surroundings.

Most of the popular scene flow methods use monocular images [1], [2] or stereo images to couple the geometry reconstruction with scene flow estimation [3], [4], [5], [6], [7], [8], [9]. However, the accuracy of such image-based solutions is still constrained by the images quality and the illumination conditions.

In contrary, LiDAR sensors provide accurate measurements of the geometry (as 3D point clouds) with ongoing developments towards increasing their density (i.e. the sensor resolution). Leveraging this potential is becoming increasingly important for the accurate computation of scene flow from point clouds.

To this end, many existing approaches [10], [11], [12], [13] focus on the 3D domain and present highly accurate scene flow with better generalization compared to the image-based modalities. Such approaches use Farthest-Point-Sampling (FPS) [14], [15], [16], [17] leading to a robust feature extraction and an accurate computation for feature similarities. However, the expensive computation of FPS decreases their capabilities to operate efficiently on dense point clouds.

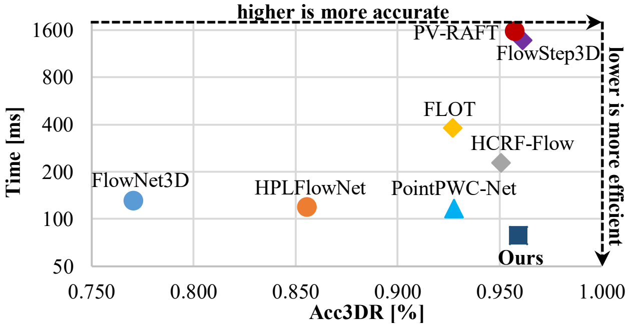

Fig. 1. Our RMS-FlowNet shows an accurate scene flow (Acc3DR) with less time-consuming. The accuracy is tested on the FlyingThings3D Subset (FT3D$_s$) [18] using 8192 points as input and the run-time is analyzed for all methods equally on a Geforce GTX 1080 Ti.

We present in this paper our RMS-FlowNet – a hierarchical point-based learning approach that relies on Random-Sampling (RS) for scene flow estimation. It is therefore more efficient, has a smaller memory footprint and shows comparable results at lower run-times compared to the state-of-the-art methods as shown in Fig. 1. The use of RS for scene flow estimation introduces big challenges and is infeasible together with existing point-based scene flow techniques [11], [12], [13]. This has mainly two reasons as clarified in Fig. 2. 1.) RS will reflect the spatial distribution of the input point cloud, which is problematic if it is far from uniform. 2.) Corresponding (rigid) areas between consecutive point clouds will be sampled differently by RS, while FPS will yield more similar patterns.

To overcome these problems, we propose a novel *Patch-to-Dilated-Patch* flow embedding, which consists of three embedding layers with lateral connections (see Fig. 5) to incorporate a larger receptive field during matching. Overall, our fully supervised architecture utilizes RS and consists of a hierarchical feature extraction, an optimized flow embedding and scene flow predictions on multiple scales. Our contribution is summarized as follows:

- We propose RMS-FlowNet – an end-to-end scene flow estimation network that operates on dense point clouds with high accuracy.
- Our network uses Random-Sampling for a hierarchical scene flow prediction in a multi-scale fashion.
- We present a novel flow embedding block (called *Patch-to-Dilated-Patch*) which is suitable for the combination with Random-Sampling.
- Exhaustive experiments show the strong results in terms of accuracy, generalization and run-time over the state-of-the-art methods.

[1] DFKI – German Research Center for Artificial Intelligence, Germany: `firstname.lastname@dfki.de`

[2] BMW Group, Germany: `Mohammad-Ali.Nikouei-Mahani@bmw.de`

II. Related Work

Learning-based scene flow from point clouds: Estimation of the scene flow from point clouds is a sub-field that became prominent with the availability of accurate LiDARs. In this domain, PointFlowNet [19] learns scene flow as a rigid motion coupled with object detection. Focusing more on point-based learning with a single flow embedding, FlowNet3D [10] proposes a learning-based architecture based on PointNet++ [15] and MeteorNet [20] adds more aspects by aggregating features from spatiotemporal neighbor points. PointPWC-Net [11] is the first point-based approach that predicts scene flow hierarchically based on [16] without structuring or ordering them. Despite of its high accuracy, the designed architecture is computationally expensive because of FPS with more memory consumption. Utilizing FPS, FlowStep3D [13] computes scene flow at the coarsest level and updates it iteratively through the Gated Recurrent Unit [21]. However, this method is computationally further expensive due to the iterative update. Unlike the aforementioned methods, our design uses RS instead of the expensive FPS over all its modules presenting superior efficiency and accurate results.

Alternatively, some structure-based learning methods are employed for scene flow estimation. In this context, Ushani et al. [22] present a real-time method by constructing occupancy grids and HPLFlowNet [23] orders the points using a permutohedral lattice. Although their efficiency, the accuracy of such methods are limited. Different from structure-based learning methods, our RMS-FlowNet relies on point-based learning and exhibits more accurate results than the aforementioned methods at lower run-time.

Some other methods [24], [25], [26] lean themselves to self-supervised category having less accuracy than our RMS-FlowNet, which is designed in a fully supervised manner.

Flow embeddings: A flow embedding is a crucial part for the computation of scene flow. It focuses on the correlation and aggregation of corresponding features across subsequent measurements to encode the spatial displacements. In this context, FlowNet3D [10] proposes a single patch-to-point embedding block by searching 64 nearest neighbors across the consecutive point cloud at a low resolution, followed by a maximum pooling and a series of propagation and refinement blocks. A patch-to-patch correlation is used by HPLFlowNet [23] for 3D point clouds through the lattice representation. Recently, PointPWC-Net [11] aggregated patch-to-patch features from unstructured point clouds based on a continuous weighting [16] which is computationally heavy. Utilizing the pyramid architecture as in [11], HCRF-Flow [27] adds a high-order conditional random fields (CRFs) [28] as a refinement module to explore both point-wise smoothness and region-wise rigidity.

Utilizing FPS, HALFlow [12] proposes a hierarchical attention mechanism for flow embedding.

Recently, FLOT [29] built a model utilizing optimal transport based on global matching [30] without the use of any sampling techniques. Inspired by RAFT [31] to construct all-pair correlation fields, FlowStep3D [13] proposes point-based and PV-RAFT [32] computes point-voxel correlation fields.

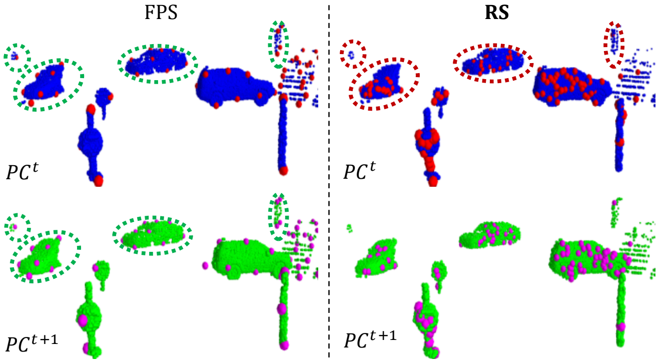

Fig. 2. The challenges of Random-Sampling (RS) (right) compared to Farthest-Point-Sampling (FPS) (left): Both techniques sample two consecutive scenes PC^t (blue) and PC^{t+1} (green) into red and pink samples, respectively.

Different from all these methods, we propose a novel and efficient *Patch-to-Dilated-Patch* flow embedding block, which works reliably together with RS without sacrificing accuracy.

III. Network Design

Our RMS-FlowNet predicts scene flow from two consecutive scans of point clouds. These point cloud sets are $PC^t = \{pc_i^t \in \mathbb{R}^3\}_{i=1}^N$ at timestamp t and $PC^{t+1} = \{pc_j^{t+1} \in \mathbb{R}^3\}_{j=1}^M$ at timestamp $t+1$, whereas (pc_i^t, pc_j^{t+1}) are the 3D Cartesian locations and (N, M) are the sizes of each set. Our network is invariant to random permutations of the point sets.

RMS-FlowNet seeks to find the similarities between point clouds to estimate the motion as scene flow vectors $SF^t = \{sf_i^t \in \mathbb{R}^3\}_{i=1}^N$ with respect to the reference view at timestamp t, i.e. sf_i^t is the motion vector for pc_i^t. The model is designed to predict the scene flow at multiple levels through hierarchical feature extraction, flow embedding, warping and scene flow estimation. The following sections describe the components of each module in detail.

A. Feature Extraction Module

The feature extraction module consists of a feature pyramid network to extract feature sets from PC^t and PC^{t+1} separately. The construction of our module involves top-down, bottom-up pathways, and lateral skip connections between them as clarified in Fig. 3.

The top-down pathway computes a hierarchy of feature sets at four scales $L = \{l\}_{k=0}^3$ from fine-to-coarse resolution, where l_0 is the full input resolution and the resolution of the down-sampled clouds are fixed as $\{\{l\}_{k=1}^3 \mid l_1 = 2048, l_2 = 728, l_3 = 320\}$. Inspired by RandLA-Net [33], which focuses on semantic segmentation, we exploit the efficient RS strategy combined with Local-Feature-Aggregation (LFA) [33]. RS has a computational complexity of $\mathcal{O}(1)$ and is therefore

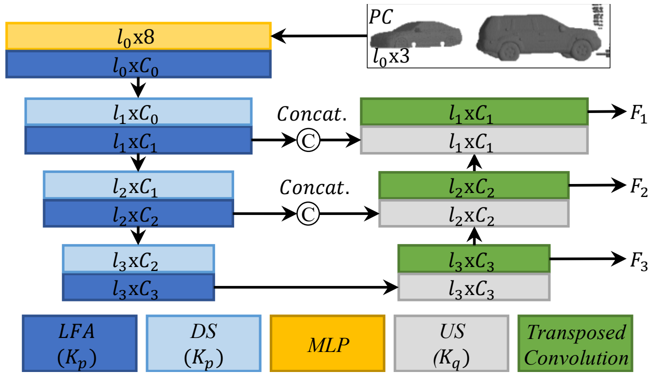

Fig. 3. Our feature extraction consists two pathways: The top-down pathway constructed of Local-Feature-Aggregation (LFA) and Down-Sampling (DS) with maximum pooling. The bottom-up pathway consists of Up-Sampling (US) and Transposed Convolution.

much more efficient compared to $\mathcal{O}(N^2)$ of FPS. Previous works [10], [11], [12], [13] take an advantage of FPS at the cost of expensive computations.

LFA is employed at all scales l_k except the finest one and starts by searching $K_p = 17$ neighbors with K-Nearest-Neighbors (KNNs) and aggregates the features with two attentive pooling layers designed as in [33]. Down-Sampling (DS) is used to reduce the resolution from l_k level to l_{k+1}. We sample randomly to the defined resolution and merge the $K_p = 17$ nearest neighbors in the higher resolution for each selected sampled with maximum pooling as shown in Fig. 3.

The bottom-up pathway in our module involves $L = \{l\}_{k=1}^{3}$ layers excluding the Up-Sampling (US) to the full input resolution. For up-scaling from level l_{k+1} to l_k, KNN is used to assign the $K_q = 1$ nearest neighbor for each point of the higher resolution to the lower one, followed by transposed convolution. To increase the quality of the features, lateral connections are added to each level. This module predicts two feature sets $F_k^t = \{f_{ki}^t \in \mathbb{R}^{C_k}\}_{i=1}^{l_k}$ and $F_k^{t+1} = \{f_{kj}^{t+1} \in \mathbb{R}^{C_k}\}_{j=1}^{l_k}$ for PC_k^t and PC_k^{t+1} respectively. Here, C_k is the feature dimension fixed as $\{\{C\}_{k=1}^{3} \mid C_1 = 128, C_2 = 256, C_3 = 512\}$. The complete feature extraction module with output channels is visualized in Fig. 3.

B. Flow Embedding

A flow embedding block across two scans is the key component for scene flow estimation. Using RS requires a special flow embedding for the mentioned difficulties in Section I. To overcome the challenges of RS, we design a flow embedding block that is different from state-of-the-art.

In this context, we establish a novel and efficient concept, called *Patch-to-Dilated-Patch*, to aggregate the relation of features. This embedding block has a larger receptive field without the need to increase the number of the nearest neighbors. To achieve this, we combine three sequential steps with lateral connections as presented in Fig. 5, and apply the entire block at each scale.

Starting by searching $K_o = 33$ nearest neighbors for each point pc_{ki}^t within PC_k^{t+1} at each scale l_k, the flow embedding consists of the following:

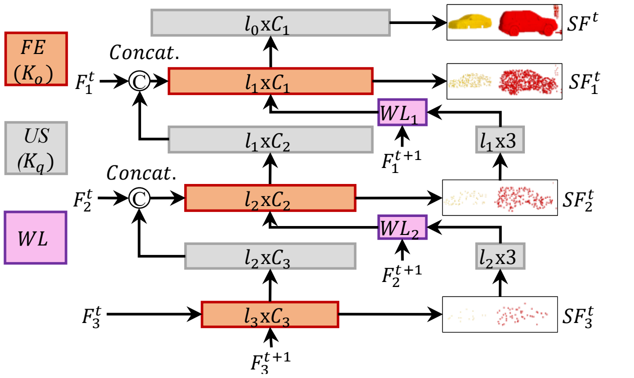

Fig. 4. Multi-scale scene flow prediction with three Flow-Embedding (FE) blocks (each consisted of three steps), two Warping-Layers (WLs), four scene flow estimators and Up-Sampling (US) blocks.

- 1st Embedding (*Patch-to-Point*): It starts by grouping K_o nearest features of F_k^{t+1} with each point pc_{ki}^t. Thereafter, these grouped features will be passed into two Multi-Layer Perceptrons (MLPs) followed by maximum pooling for feature aggregation. Each MLP yields features of C_k dimensions at scale l_k.
- 2nd Embedding (*Point-to-Patch*): It aggregates the K_p nearest features within the reference point cloud into each pc_{ki}^t by computing attention scores and summation, i.e. the features are weighted.
- 3rd Embedding (*Point-to-Dilated-Patch*): It repeats the previous step on the previous result with new attention scores for the K_p nearest features. This embedding layer results in an increased receptive field.

Technically, we do not increase the number of the nearest neighbors for the 3rd Embedding, but we aggregate features from a larger area by repeating the aggregation mechanism (see Fig. 5). Overall, the three steps result in our novel *Patch-to-Dilated-Patch* embedding. This way, we are able to obtain a larger receptive field with a small number of nearest neighbors, which is computationally more efficient.

The attention-based aggregation technique [34], [35] learns attention scores for each embedded feature f_{ki}^t, followed by a *softmax* to suppress the least correlated features. Then, the features are weighted by the attention scores and summed up.

Additionally, we concatenate the features F_k^t and add a residual connection (Res. Conn.) to increase the quality of our flow embedding (c.f. Fig. 5). This design is validated in the ablation study (Section IV-E).

C. Multi-Scale Scene Flow Estimation

As mentioned, RMS-FlowNet predicts scene flow at multiple scales inspired by PointPWC-Net [11], but we consider significant changes in the conjunction with RS to make our prediction more efficient. Our prediction of scene flow over all scales consists of two Warping-Layers (WLs), three Flow-Embeddings (FEs), three scene flow estimators and Up-Sampling (US) blocks as shown in Fig. 4. Compared to the design of PointPWC-Net [11], we save one element from

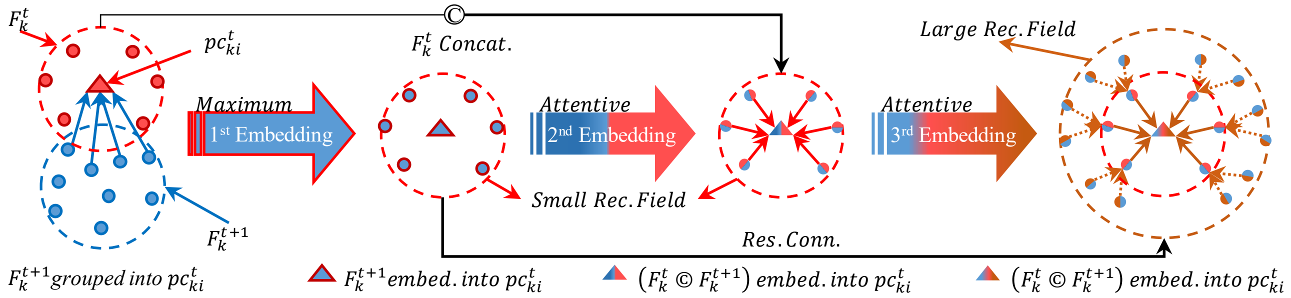
F_k^t
pc_{ki}^t
F_k^t Concat.
Large Rec. Field
Maximum
1st Embedding
Attentive
2nd Embedding
Attentive
3rd Embedding
Small Rec. Field
F_k^{t+1}
Res. Conn.
F_k^{t+1} grouped into pc_{ki}^t
F_k^{t+1} embed. into pc_{ki}^t
$(F_k^t \copyright F_k^{t+1})$ embed. into pc_{ki}^t
$(F_k^t \copyright F_k^{t+1})$ embed. into pc_{ki}^t

Fig. 5. Our novel Flow-Embedding (FE) block consists of three main steps: A maximum embedding across point clouds followed by two attentive embedding layers. It further uses lateral connections: A feature concatenation (F_k^t *Concat.*) and a residual connection (*Res. Conn.*).

each category and we base our complete design attention mechanisms. Consequentially, we speed-up our model without sacrificing any accuracy as shown in our results, c.f. in Table I. The multi-scale estimation starts at the coarsest resolution by prediction SF_3^t with a scene flow estimation module after a first FE. The estimation module consists of just three MLPs with 64, 32 and 3 output channels, respectively. Thereafter, we up-sample the estimated scene flow as well as the coming features from the FE to the next higher scale using KNN with $K_q = 1$.

Our Warping-Layer utilizes the up-sampled scene flow SF_k^t at scale level l_k to warp F_k^{t+1} towards F_k^t. To this end, we add the predicted scene flow SF_k^t to PC_k^t to compute the warped $\widetilde{PC}_k^{t+1}$ and then we group the features F_k^{t+1} into F_k^t by using KNN search across PC_k^{t+1} and $\widetilde{PC}_k^{t+1}$. This warping is more simple and efficient compared to the process in PointPWC-Net [11] which associates firstly the predicted scene flow to PC_k^{t+1} by KNN search in order to warp PC_k^{t+1} into $\widetilde{PC}_k^t$ and then grouping the features with another KNN search.

D. Loss Function

The model is a fully supervised at multiple scales, similar to PointPWC-Net [11]. If SF_k^t is the predicted scene flow and the ground truth is $SF_{GT,k}^t$ at level l_k, then the objective can be written as:

$$\mathscr{L}(\theta) = \sum_{k=0}^{3} \alpha_k \sum_{i=1}^{l_k} \|sf_{ki}^t - sf_{GT,ki}^t\|_2, \quad (1)$$

with $\|.\|_2$ denoting the L_2-norm and weights per scale of $\{\{\alpha_k\}_{k=0}^3 \mid \alpha_0 = 0.02, \alpha_1 = 0.04, \alpha_2 = 0.08, \alpha_3 = 0.16\}$.

IV. Experiments

We run several experiments to evaluate the results of our RMS-FlowNet for scene flow estimation. Firstly, we demonstrate the accuracy and the efficiency of RMS-FlowNet compared to the state-of-the-art. Secondly, we verify our design choice with several analyses.

A. Evaluation Metrics and Data Sets

For a fair comparison, we use the same evaluation metrics as in [23]. Let SF^t denote the predicted scene flow, and SF_{GT}^t the ground truth scene flow. The evaluation metrics are averaged over all points and computed as follows:

- *EPE3D [m]*: The 3D end-point error computed in meters as $\|SF^t - SF_{GT}^t\|_2$.
- *Acc3DS [%]*: The strict 3D accuracy which is the ratio of points whose EPE3D < 0.05 m **or** relative error $<$ 5%.
- *Acc3DR [%]*: The relaxed 3D accuracy which is the ratio of points whose EPE3D < 0.1 m **or** relative error $<$ 10%.
- *Out3D [%]*: The ratio of outliers whose EPE3D > 0.3 m **or** relative error $>$ 10%.

We train our RMS-FlowNet on the established data set FlyingThings3D Subset (FT3D$_s$) [18] which consists of 19640 labeled scene flow scenes available in the training set. We exclude the occluded points and the points with a depth above 35 meters as [10], [11], [12], [13], [23], [29], [32] considering most of the moving objects within the scenes.

For testing, we evaluate our model on all 3824 frames available in the test split of FT3D$_s$. Since FT3D$_s$ scenes are only synthetic data, we verify the generalization ability of our model to real-world scenes of the KITTI [36] data set without fine-tuning. For both data sets FT3D$_s$ and KITTI, the setup of evaluation is exactly the same as in related works.

Since the existing labeled data does not provide a direct representation of point cloud information (i.e. 3D Cartesian locations), we follow the established pre-processing strategy of HPLFlowNet [23][1] which is commonly used also in the state-of-the-art methods.

For training and evaluation with a specific resolution, the pre-processed data is randomly sub-sampled to N points with random order.

B. Implementation and Training

We use the Adam optimizer with default parameters and train our model with 800 epochs divided in two phases: To speed up the convergence of our model, we first train 120 epochs with a fixed set of points for each frame and apply an exponentially decaying learning rate, initialized with 0.001, then decreased with a decaying rate of 0.7 each 10 epochs. For the next 680 epochs, the learning rate is fixed to 0.0001 and 8192 points are sampled randomly for each frame in each iteration.

[1] `https://github.com/laoreja/HPLFlowNet.`

TABLE I

Quantitative results of our RMS-FlowNet compared to the state-of-the-art methods on FT3D$_s$ [18] and KITTI [36]. Our RMS-FlowNet is trained and tested on 8192 points as other methods. The scores of the state-of-the-art methods are obtained from [10], [11], [12], [13], [23], [29], [32]. The best and the second best scores in all metrics are emboldened and underlined respectively. The run-time and the memory usage are compared equally on a Geforce GTX 1080 Ti. We mention the sampling strategy of each method to facilitate the comparison. Previous work either applies lattice-based scaling, avoids sampling, or applies Farthest-Point-Sampling (FPS). Our RMS-FlowNet uses Random-Sampling (RS) and shows robust and accurate results at a low run-time.

Data set	Model	Sampling	EPE3D [m]	Out3D [%]	Acc3DS [%]	Acc3DR [%]	Time [ms]	Memory [GB]
FT3D$_s$ [18]	FlowNet3D [10]	FPS	0.114	0.602	0.413	0.771	132	10.85
	HPLFlowNet [23]	Scaling	0.080	0.428	0.616	0.856	119	1.58
	PointPWC-Net [11]	FPS	0.059	0.342	0.738	0.928	117	2.86
	FLOT [29]	-	0.052	0.357	0.732	0.927	376	3.84
	HALFlow [12]	FPS	0.049	0.308	0.785	0.947	-	-
	FlowStep3D [13]	FPS	**0.046**	**0.217**	0.816	**0.961**	1369	**1.31**
	PV-RAFT [32]	-	**0.046**	0.292	**0.817**	0.957	1565	4.03
	RMS-FlowNet (Ours)	**RS**	0.056	0.324	0.792	0.955	**77**	1.39
KITTI [36]	FlowNet3D [10]	FPS	0.177	0.527	0.374	0.668	132	10.85
	HPLFlowNet [23]	Scaling	0.117	0.410	0.478	0.778	119	1.58
	PointPWC-Net [11]	FPS	0.069	0.265	0.728	0.888	117	2.86
	FLOT [29]	-	0.056	0.242	0.755	0.908	376	3.84
	HALFlow [12]	FPS	0.062	0.249	0.765	0.903	-	-
	FlowStep3D [13]	FPS	0.055	**0.149**	0.805	0.925	1369	**1.31**
	PV-RAFT [32]	-	0.056	0.216	**0.823**	0.937	1565	4.03
	RMS-FlowNet (Ours)	**RS**	**0.053**	0.203	0.818	**0.938**	**77**	1.39

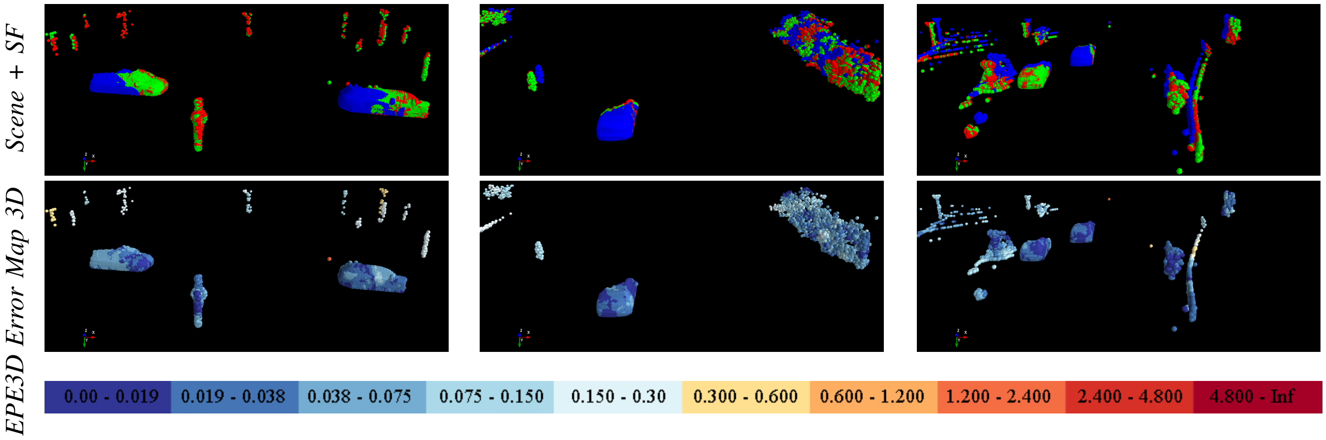

Fig. 6. Three examples taken from KITTI show the impact of our RMS-FlowNet. The first row of each example visualizes PC^t as blue color and the predicted and ground truth scene flow after adding them to PC^t in red and green color, respectively. The second row shows the end-point error in meters for each scene according to the color map shown in the last row. Our RMS-FlowNet shows low errors (dark blue) across a wide range of the observed scene especially for moving objects (cars).

Moreover, we add geometrical augmentation as in the related works, i.e. points are randomly rotated around the X, Y and Z axes by a small angle and a random translational offset is added to increase the ability of our model to generalize without fine-tuning.

C. Comparison to State-of-the-Art

Evaluation on FT3D$_s$: In order to demonstrate the accuracy, generalization and efficiency of our model, we compare to the state-of-the-art methods in Table I. Our RMS-FlowNet outperforms the methods of [10], [11], [23] over all evaluation metrics and shows comparable results to [12], [13], [29], [32] with lowest run-time and low memory footprint. Compared to the concurrent methods [12], [13], which use FPS, our RMS-FlowNet shows comparable accuracy utilizing RS.

Generalization to KITTI: We test the generalization ability to the KITTI data set [36] without fine-tuning. The reported scores in Table I provide an evidence about the robustness on real-world scenes. Our RMS-FlowNet outperforms over all the methods of [10], [11], [12], [23], [29] and presents comparable results to [13], [32].

Visually, three examples on KITTI are shown in Fig. 6 where the scene flow of a moving car and the surroundings are with low deviations compared to ground truth.

Efficiency: To verify the efficiency of RMS-FlowNet, we run the official implementations of the state-of-the-art meth-

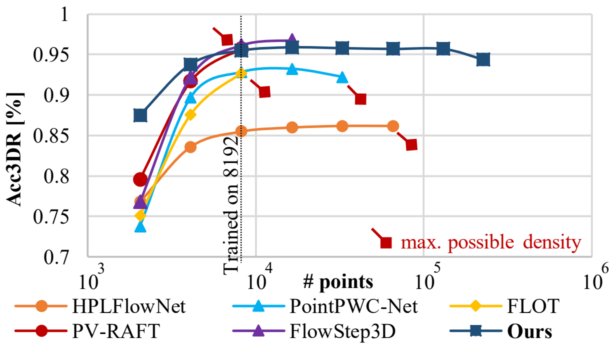

Fig. 7. Analysis of the accuracy for different numbers of points compared to state-of-the-art methods.

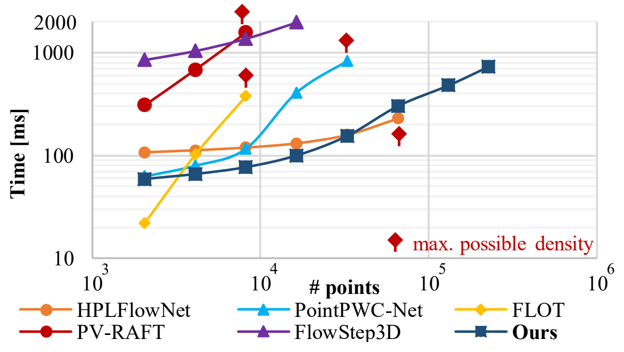

Fig. 8. Run-time analysis for different numbers of points compared to state-of-the-art methods.

ods [10], [11], [13], [23], [29], [32] on a clean environment with a GeForce GTX 1080 Ti and measure the average inference time in milliseconds (ms) over the test set. As shown in Table I, RMS-FlowNet is more efficient in terms of run-time than any other method for 8192 input points and it's near to [13] in terms of memory use. Hence, our method is ~1.5x faster than [10], [11], [23], ~4.5x faster than FLOT [29] and ~18x faster than [13], [32] which are the main competitor to our method in terms of accuracy.

Due to the unavailability of an open source code of HALFlow [12] and missing efficiency analysis in its original paper, we can not analyze the efficiency but we assume its less efficiency than us due to the use of FPS.

D. Varying Point Densities

We evaluate our method against the important competitors [11], [13], [23], [29], [32] on different point densities as shown in Fig. 7. Acc3DR and inference time on FT3D$_s$ are measured for a wide range of densities $N = \{2048 * 2^i\}_{i=0}^{6}$, and finally all available non-occluded points are used, which corresponds on average to ~225K points (see Fig. 7 and Fig. 8). For the competing methods of FLOT [29], PV-RAFT [32], PointPWC-Net [11] and HPLFlowNet [23], the maximum possible densities are limited to 8192, 8192, 32768 and 65536, respectively, due to exceeding the memory limit of the Geforce GTX 1080 Ti for our tested range. We have limited as well the number of points for FlowStep3D [13] to 16384 in our test, due to the bad run-time ($>$ 2.5 seconds) for each frame with more dense scenes.

In contrast, RMS-FlowNet can operate on more than 250K points efficiently with high accuracy and low run-time. In order to keep the accuracy stable for densities $>$ 32K, we have increased the resolution of the down-sampled features (c.f. Section III-A) to $\{\{l\}_{k=1}^{3} \mid l_1 = 8192, l_2 = 2048, l_3 = 512\}$ without further training or fine-tuning at all. As a result, the accuracy remains stable over a wide range of densities (see Fig. 7). Even with this change, our RMS-FlowNet is more efficient and faster than previous works for increased input densities (see Fig. 8). The design of RMS-FlowNet allows to operate on a much higher maximum density compared to other methods, due to their memory footprint and the time consumption. However, the run-time of our RMS-FlowNet still increases super-linear with growing input density due to the KNN search.

E. Ablation Study

Finally, we verify our design choices for the FE on FT3D$_s$ [18] by removing components of the FE and compare the variants in Table II. The models in this comparison are trained only for the first phase as explained in Section IV-B and without augmentation. Each part clearly adds a contribution to the overall accuracy when using RS. The 1st Embedding with maximum pooling, as basically used in FlowNet3D [10], is not able to resolve the challenges of the RS strategy for scene flow estimation. The complete design – three embedding layers with lateral connections – leads to the best results.

TABLE II

ABLATION STUDY ON VARIOUS DESIGN VARIANTS FOR THE FLOW EMBEDDING OF OUR RMS-FLOWNET. WE TEST ALL VARIANTS ON 8192 POINTS FROM THE TEST SPLIT OF FT3D$_s$ [18].

1st Embed.	2nd Embed.	F_k^t Conn.	Res. Concat.	3rd Embed.	FT3D$_s$ [18] Acc3DR [%]
✓	✗	✗	✗	✗	0.792
✓	✓	✗	✗	✗	0.860
✓	✓	✓	✗	✗	0.871
✓	✓	✓	✓	✗	0.885
✓	✓	✓	✓	✓	**0.897**
w/ full training and augmentation					**0.955**

V. Conclusion

In this paper, we have proposed RMS-FlowNet – an efficient and fully supervised network for multi-scale scene flow estimation in large-scale point clouds. Utilizing Random-Sampling (RS) during feature extraction, we could boost the run-time and memory footprint to allow for an efficient processing of point clouds at an unmatched maximum density. The novel Flow-Embedding block (called *Patch-to-Dilated-Patch*), resolves the prominent challenges when using RS for scene flow estimation. Consequentially, RMS-FlowNet reaches state-of-the-art accuracy on FT3D$_s$ and generalizes well over a wide range of input densities as well as to the real-world scenes of KITTI.

Acknowledgment

This work was partially funded by the BMW Group and partially by the Federal Ministry of Education and Research Germany under the project DECODE (01IW21001).

REFERENCES

[1] F. Brickwedde, S. Abraham, and R. Mester, "Mono-SF: Multi-View Geometry Meets Single-View Depth for Monocular Scene Flow Estimation of Dynamic Traffic Scenes," in *IEEE/CVF International Conference on Computer Vision (CVPR)*, 2019.

[2] J. Hur and S. Roth, "Self-Supervised Monocular Scene Flow Estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[3] F. Aleotti, M. Poggi, F. Tosi, and S. Mattoccia, "Learning End-To-End Scene Flow by Distilling Single Tasks Knowledge," in *Conference on Artificial Intelligence (AAAI)*, 2020.

[4] E. Ilg, T. Saikia, M. Keuper, and T. Brox, "Occlusions, Motion and Depth Boundaries with a Generic Network for Disparity, Optical Flow or Scene Flow Estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[5] H. Jiang, D. Sun, V. Jampani, Z. Lv, E. Learned-Miller, and J. Kautz, "SENSE: a Shared Encoder Network for Scene-flow Estimation," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[6] R. Saxena, R. Schuster, O. Wasenmüller, and D. Stricker, "PWOC-3D: Deep Occlusion-Aware End-to-End Scene Flow Estimation," *IEEE International Conference on Intelligent Vehicles Symposium (IV)*, 2019.

[7] R. Schuster, C. Unger, and D. Stricker, "A Deep Temporal Fusion Framework for Scene Flow Using a Learnable Motion Model and Occlusions," in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021.

[8] R. Schuster, O. Wasenmüller, G. Kuschk, C. Bailer, and D. Stricker, "SceneFlowFields: Dense Interpolation of Sparse Scene Flow Correspondences," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[9] G. Yang and D. Ramanan, "Upgrading Optical Flow to 3D Scene Flow through Optical Expansion," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[10] X. Liu, C. R. Qi, and L. J. Guibas, "FlowNet3D: Learning Scene Flow in 3D Point Clouds," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[11] W. Wu, Z. Y. Wang, Z. Li, W. Liu, and L. Fuxin, "PointPWC-Net: Cost Volume on Point Clouds for (Self-) Supervised Scene Flow Estimation," in *European Conference on Computer Vision (ECCV)*, 2020.

[12] G. Wang, X. Wu, Z. Liu, and H. Wang, "Hierarchical Attention Learning of Scene Flow in 3D Point Clouds," *IEEE Transactions on Image Processing (TIP)*, 2021.

[13] Y. Kittenplon, Y. C. Eldar, and D. Raviv, "FlowStep3D: Model Unrolling for Self-Supervised Scene Flow Estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[14] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution On χ-Transformed Points," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS)*, 2018.

[15] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems (NIPS)*, 2017.

[16] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep Convolutional Networks on 3D Point Clouds," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[17] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, "PointWeb: Enhancing Local Neighborhood Features for Point Cloud Processing," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[18] 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 *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[19] A. Behl, D. Paschalidou, S. Donné, and A. Geiger, "PointFlowNet: Learning Representations for Rigid Motion Estimation from Point Clouds," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[20] X. Liu, M. Yan, and J. Bohg, "MeteorNet: Deep Learning on Dynamic 3D Point Cloud Sequences," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[21] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[22] A. K. Ushani and R. M. Eustice, "Feature Learning for Scene Flow Estimation from LIDAR," in *Conference on Robot Learning (CoRL)*, 2018.

[23] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[24] R. Li, G. Lin, and L. Xie, "Self-Point-Flow: Self-Supervised Scene Flow Estimation from Point Clouds with Optimal Transport and Random Walk," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[25] J. K. Pontes, J. Hays, and S. Lucey, "Scene flow from point clouds with or without learning," in *International Conference on 3D Vision (3DV)*, 2020.

[26] H. Mittal, B. Okorn, and D. Held, "Just Go with the Flow: Self-Supervised Scene Flow Estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)n*, 2020.

[27] R. Li, G. Lin, T. He, F. Liu, and C. Shen, "HCRF-Flow: Scene Flow from Point Clouds with Continuous High-order CRFs and Position-aware Flow Embedding," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[28] K. Ristovski, V. Radosavljevic, S. Vucetic, and Z. Obradovic, "Continuous Conditional Random Fields for Efficient Regression in Large Fully Connected Graphs," in *Conference on Artificial Intelligence (AAAI)*, 2013.

[29] G. Puy, A. Boulch, and R. Marlet, "FLOT: Scene Flow on Point Clouds Guided by Optimal Transport," in *European Conference on Computer Vision (ECCV)*, 2020.

[30] V. Titouan, N. Courty, R. Tavenard, and R. Flamary, "Optimal Transport for structured data with application on graphs," in *International Conference on Machine Learning*, 2019.

[31] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *European conference on computer vision (ECCV)*, 2020.

[32] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[33] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[34] B. Yang, S. Wang, A. Markham, and N. Trigoni, "Robust Attentional Aggregation of Deep Feature Sets for Multi-view 3D Reconstruction," *International Journal of Computer Vision (IJCV)*, 2020.

[35] W. Zhang and C. Xiao, "PCAN: 3D Attention Map Learning Using Contextual Information for Point Cloud Based Retrieval," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[36] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.