SSRFlow: Semantic-aware Fusion with Spatial Temporal Re-embedding for Real-world Scene Flow

Supplementary Material

7. Network

7.1. Network Architecture

Our proposed network architecture mainly consists of three components: (1) Hierarchical point cloud feature extraction. (2) Global attentive flow initialization. (3) Local flow refinement. To hierarchically extract semantic features from point clouds, we begin by adopting a pyramid feature extraction network. Subsequently, we construct an attentive global flow embedding that accounts for both high-dimensional feature space and Euclidean space. The resulting sparse flow is then upsampled using upsampling and warping layers, enabling denser flows at lower levels. These flows are accumulated onto the source frame, ultimately yielding the warped source frame. Thereafter, spatiotemporal re-embedding features are used for hierarchical refinement of the residual scene flow between the warped source frame and the target frame. Through iterative refinement, full-resolution scene flow is generated as the ultimate result.

The architecture of our model comprises a total of 5 levels. We take $N_1 = M_1 = 8192$ points as input, randomly sampled from source and target frames if they contain different numbers of points. Within the multi-layer pyramid network, the down-sampled points vary across different levels, with $N_2 = 2048$, $N_3 = 512$, $N_4 = 256$, and $N_5 = 64$, respectively. Notably, the $5th$ layer is referred to as the GF module.

7.2. Local Flow Embedding

In this section, we input the warped source frame and target frame into the Local Flow Embedding (LFE) module to generate a local flow embedding, which is realized following the patch-to-patch approach [40].

The matching cost between two frames of point clouds can be defined as follows:

$$cost\left(y_j, wx_k\right) = \text{MLP}(\eta(STRF_k, g_j, y_j - wx_k)), \quad (16)$$

where η stands for concatenation. $STRF_k$ is the feature of warped source point wx_k, and g_j is the feature of target point y_j. Afterward, this matching cost can be aggregated into a point-to-patch cost volume between the two frames.

$$CV_{point}(wx_k) = \sum_{y_j \in \mathcal{N}_T(wx_k)} \text{MLP}(y_j - wx_k) cost\left(y_j, wx_k\right). \quad (17)$$

Finally, the patch-to-patch cost volume for warped source point wx_i can be defined as:

$$CV_{patch}\left(wx_i\right) = \sum_{wx_k \in \mathcal{N}_{WS}(wx_i)} \text{MLP}(wx_k - wx_i) CV_{point}(wx_k). \quad (18)$$

Here $\text{MLP}(wx_k - wx_i)$ and $\text{MLP}(y_j - wx_k)$ are the aggregation weights determined by the direction vectors. Follow the notation in the main text, we use $\mathcal{N}_{WS}(wx_i)$ to define the neighbors of wx_i in frame WS and $\mathcal{N}_T(wx_k)$ to define the neighbors of wx_k in frame T. The patch-to-patch manner employed in the cost volume is illustrated in Figure 8.

8. Experiments Settings

8.1. Evaluation Metrics

For fair comparisons, we leverage the same set of evaluation metrics as in the previous methods [4, 21, 33, 40].

- EPE3D: the main evaluation measuring average 3D end-point-error which is formulated as $\text{EPE3D} = \frac{1}{N}\sum_i \|\tilde{sf}_i - sf_i\|_2$
- AS3D: the strict version for scene flow accuracy which denotes the percentage of points whose EPE3D < 0.05m or relative error $< 5\%$.
- AR3D: the relax version for scene flow accuracy which denotes the percentage of points whose EPE3D < 0.1m or relative error $< 10\%$.
- Out3D: the proportion of points that exhibit significant miscalculations, characterized by an EPE3D > 0.3m or relative error $> 30\%$.
- EPE2D: the main measurement for 2D optical flow which represents the end-point-error by projecting points black to the 2D image plane.
- Acc2D: the percentage of points whose EPE2D < 3px or relative error $< 5\%$.

8.2. Search Methods

To assess the enhancement of the KNN+Radius approach compared to the pure KNN method, we visualize the flow disparities of different points under the search strategies of KNN and KNN+Radius local point clusters based on the ground truth of the scene flow.

From Figure 9 it can be observed that using only KNN introduces noise points that do not belong to the block, thereby causing greater local flow disparities and leading to deviating in the network model learning.

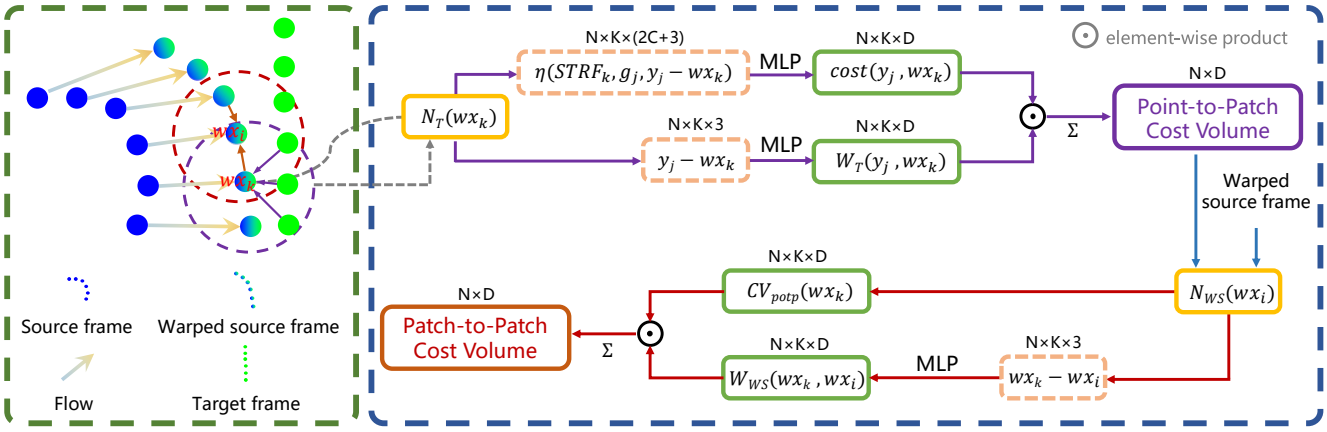
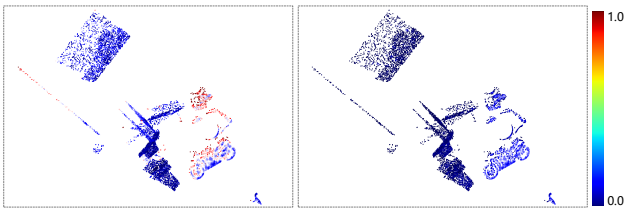


Figure 8. The structure of the Local Flow Embedding (LFE) module. We first generate $\mathcal{N}_{WS}(wx)$ and $\mathcal{N}_T(wx)$ using KNN for each warped source point wx. Then, we input each point and its neighbors from frames WS and T along with their features into the LFE module.The spatiotemporal re-embedded feature of the warped point wx_k is combined with the features of its neighboring points in $\mathcal{N}_T(wx_k)$, along with the direction vector $y_j - wx_k$. Subsequently, they are processed through an MLP to calculate a point-to-patch cost between wx_k and the target frame T. Following this, each point-to-patch cost in $\mathcal{N}_{WS}(wx_i)$ is additionally self-aggregated using an MLP to construct a patch-to-patch cost volume.



(a) KNN (b) KNN+Radius

Figure 9. The differences of ground truth scene flow local consistency under different nearest point search methods. The normalized value is shown in the right color bar.

8.3. Hyper-parameters in DA Losses

We conduct a comprehensive experimental analysis on the hyper-parameters $K, R,$ and TH in the context of DA Losses, where R represents the radius truncation of neighbor search, K represents the number of neighbors in KNN, and TH represents the cross-frame feature similarity threshold. It is worth mentioning that R and K are uniform across both Local Flow Consistency (LCF) and Cross-frame Feature Similarity (CFS) loss functions since they are utilized to identify local rigid blocks with similar semantic features.

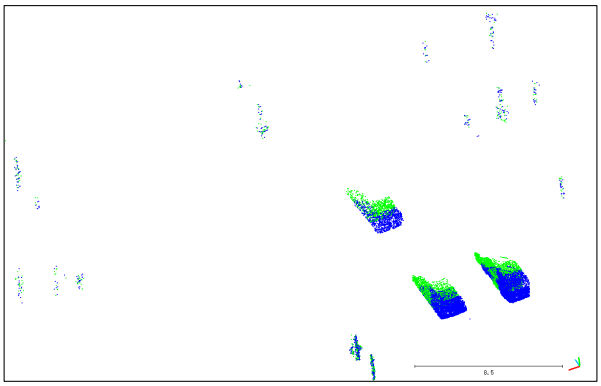
To analyze the local consistency of ground truth (GT) flow, we initiate the process by selecting a suitable scenario in the KITTIs dataset that exhibits minimal visible background points, as depicted in Figure 10. Subsequently, we systematically augment the ranges of R and K to acquire diverse consistency outcomes. As shown in Figure 10 and Figure10, the quantity of KNN group points exhibits a gradual decrease as the truncation radius decreases. Simultaneously, the average flow differences within each group also decrease, indicating that the KNN+Radius search method effectively eliminates the noise points.

We further explore the disparities between KNN and KNN+Radius search methods. The KNN+Radius search method results in certain points without any neighboring points. These isolated points, known as exceptionally sparse or invalid points, function as noise points that impede flow smoothing. However, utilizing the pure KNN search method would optimize flow consistency between noise points and mandatory surrounding points, which is an incorrect outcome. By observing the size of local rigid blocks in real-world scenarios, we finally select $R = 0.05$m as the truncation radius to perform ablation experiments on different KNN search numbers. The results are shown in Figure 10.

Threshold TH	0.99	0.95	0.9	0.8	0.7	0.6
EPE3D	0.0149	0.0114	0.0122	0.0138	0.0151	0.0180

Table 8. The EPE3D of KITTIs with different cross-frame feature similarity threshold.

Regarding the selection of hyper-parameter TH in the CFS loss function, we fix $K = 32$ and $R = 0.05$m that represent a local rigid block for capturing cross-frame similar features and test different TH values in the ablation experiments. The corresponding results are listed in Table 8. Our model was trained for a limited number of **450 epochs** on the FT3Ds dataset and subsequently evaluated on the KITTIs dataset without undergoing any fine-tuning, which is enough for observing the trend.



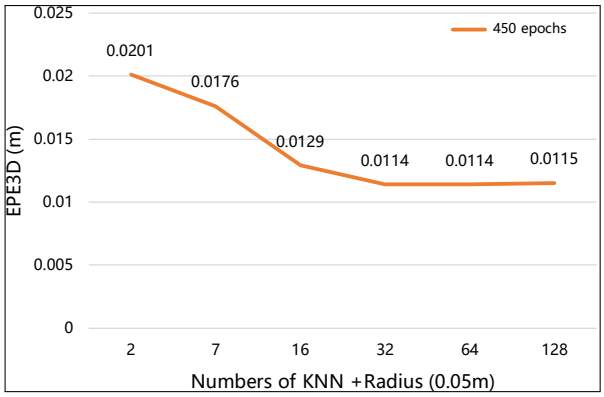
(a) The visualization of a KITTIs dataset scene in which there are rare background points observed.

$R\ (m) \setminus K$	2	7	16	32	64	128
0.001	0.00102	0.00104	0.00104	0.00104	0.00104	0.00104
0.0025	0.00106	0.00111	0.00111	0.00111	0.00111	0.00111
0.005	0.00108	0.00113	0.00114	0.00114	0.00114	0.00114
0.0075	0.00108	0.00113	0.00115	0.00116	0.00116	0.00116
0.01	0.00108	0.00113	0.00115	0.00117	0.00117	0.00117
0.015	0.00108	0.00114	0.00116	0.00118	0.00119	0.00119
0.02	0.00108	0.00114	0.00117	0.00119	0.00121	0.00121
0.05	0.00107	0.00113	0.00116	0.0012	0.00124	0.00128
0.1	0.00106	0.00112	0.00116	0.0012	0.00125	0.00132
0.5	0.00106	0.00112	0.00116	0.0012	0.00126	0.00135
0.75	0.00106	0.00112	0.00116	0.0012	0.00126	0.00135
1	0.00106	0.00112	0.00116	0.0012	0.00126	0.00135
2	0.00106	0.00112	0.00116	0.0012	0.00126	0.00135
5	0.00118	0.00124	0.00128	0.00132	0.00138	0.00147
∞	0.00118	0.00124	0.00128	0.00232	0.00519	0.007

(b) A color level plot to visualize the average local differences in GT flow for different combinations of R and K.

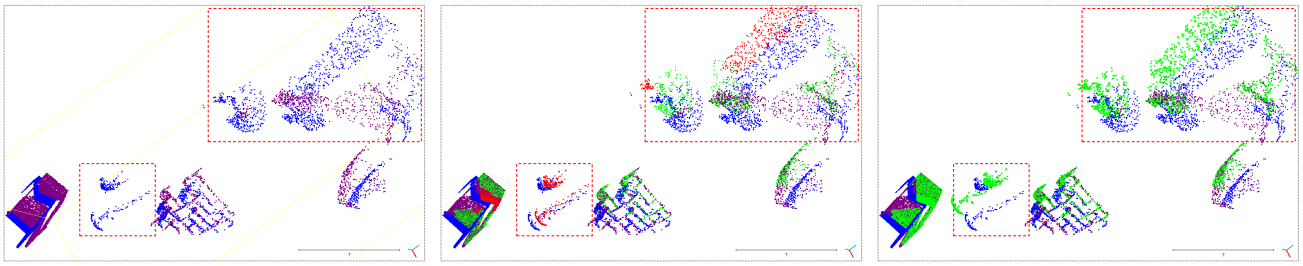
$R\ (m) \setminus K$	2	7	16	32	64	128
0.001	62.33%	27.51%	12.12%	6.06%	3.03%	1.51%
0.0025	83.36%	58.33%	30.08%	15.06%	7.53%	3.76%
0.005	91.95%	78.96%	56.76%	30.87%	15.43%	7.71%
0.0075	94.36%	86.79%	70.58%	46.24%	23.32%	11.66%
0.01	95.53%	89.98%	77.85%	57.88%	31.06%	15.53%
0.015	96.50%	92.91%	85.72%	71.26%	46.40%	23.26%
0.02	97.13%	94.17%	89.52%	78.64%	58.56%	31.05%
0.05	98.54%	96.58%	94.56%	91.50%	82.45%	65.60%
0.1	99.34%	97.81%	96.15%	94.49%	91.07%	81.74%
0.5	99.95%	99.66%	98.48%	97.11%	96.07%	94.25%
0.75	99.95%	99.78%	98.89%	97.44%	96.47%	95.09%
1	99.97%	99.84%	99.12%	97.68%	96.64%	95.54%
2	99.99%	99.97%	99.59%	98.40%	97.12%	96.25%
5	100.00%	99.99%	99.84%	99.21%	97.96%	96.84%
∞	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

(c) As the value of R progressively increases, a color level plot illustrates the proportion of points accessible through KNN + Radius compared to the original KNN. As the radius approaches infinity, the original KNN behavior is reinstated.



(d) The EPE3D of KITTIs with increasing the number of KNN searches on a reasonable local rigid radius ($R = 0.05$ m). Notably, our model is tested on the KITTIs dataset without any fine-tuning.

Figure 10. Ablation studies and analysis of adaption losses.



(a) (b) (c)

Figure 11. (a) The occlusion occurs between the source frame and the target frame. In this scenario, red bounding boxes delineate points in the source frame that vanish when transitioning to the target frame. (b) The green points represent the result of warping the source frame using our predictions. Here, the red points indicate incorrectly predicted warped points that have an EPE3D greater than 0.1m. (c) The green points depict the result of warping the source frame using ground truth.

9. Limitations

Inferring motion flow for occluded objects has always been a challenging task. Particularly, in situations where objects are completely occluded, as defined in [27], motion inference relying on the consistency of local motion becomes implausible. In real-world scenarios, taking into account roadway regulations, it might be feasible to estimate the movement of occluded vehicles on the lane by considering the overall directionality of traffic flow. However, for synthetic datasets like FT3Do, where random motion is assigned, it is challenging to infer the motion of completely occluded objects even

Model	EPE3D↓	Param size (M)	Run time (ms)
PointPWC	0.0588	7.72M	76ms
PointPWC+STR	0.0504	8.02M	81ms
PointPWC+STR+GF	**0.0402**	9.89M	96ms
Bi-PointflowNet	0.0282	7.96M	80ms
Bi-PointFlow+GF	0.0227	9.21M	87ms
Bi-PointFlow+GF+STR	**0.0187**	10.85M	103ms
FlowNet3D	0.1136	1.23M	70ms
FlowNet3D+GF	**0.0837**	1.36M	94ms
WM3DSF	0.0281	4.77M	63ms
WM3DSF+DCA Fusion	**0.0209**	5.37M	72ms

Table 9. Transfer results on FT3Ds.

for humans due to the unpredictable and random nature of the motion generation. Our model exhibits poor performance in some completely occluded synthetic scenes, as illustrated in Figure 11. An effective approach involves integrating a specific pattern of motion (object-wise relation) during the generation of a synthesized scene flow dataset.

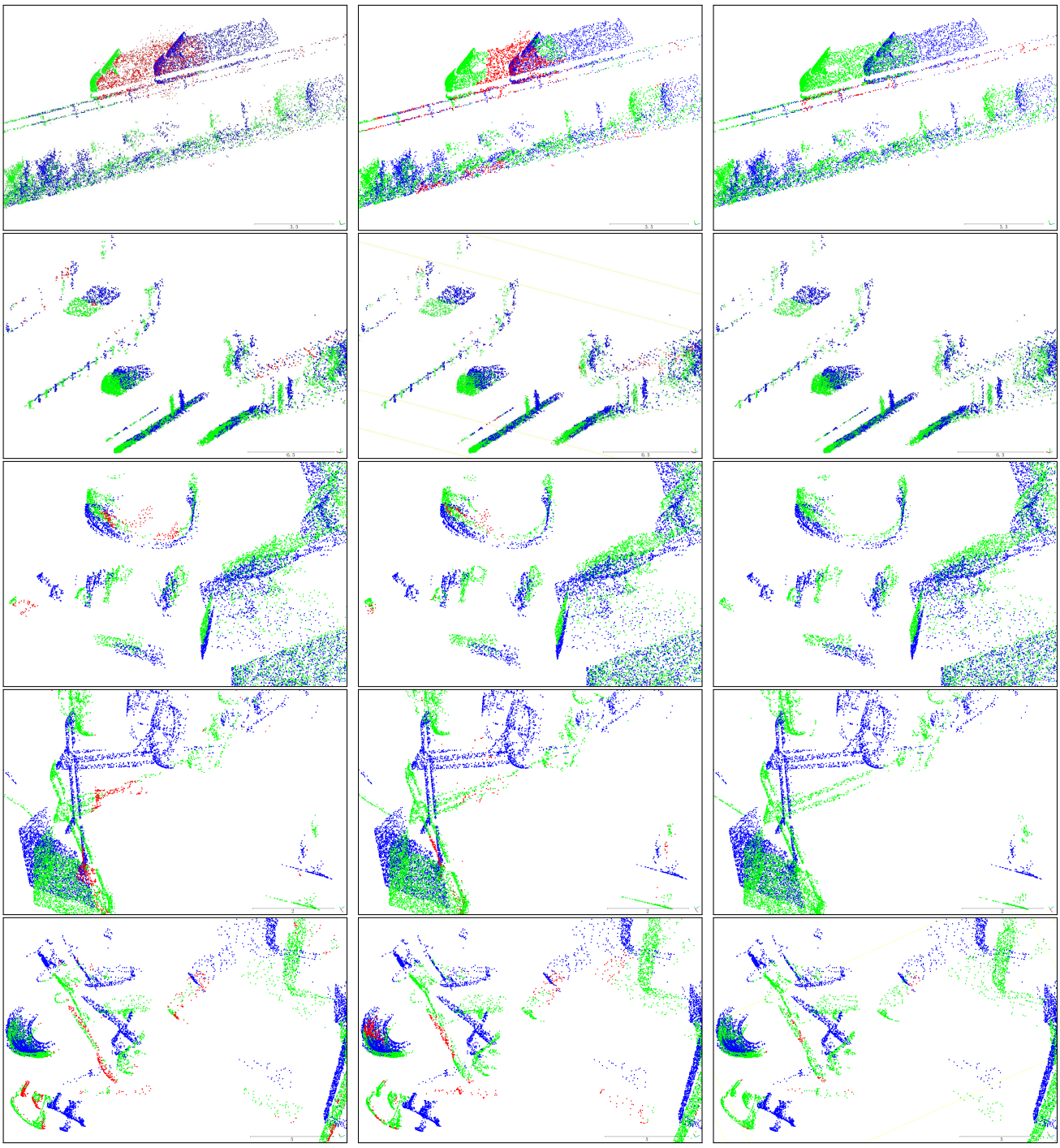
10. Datasets

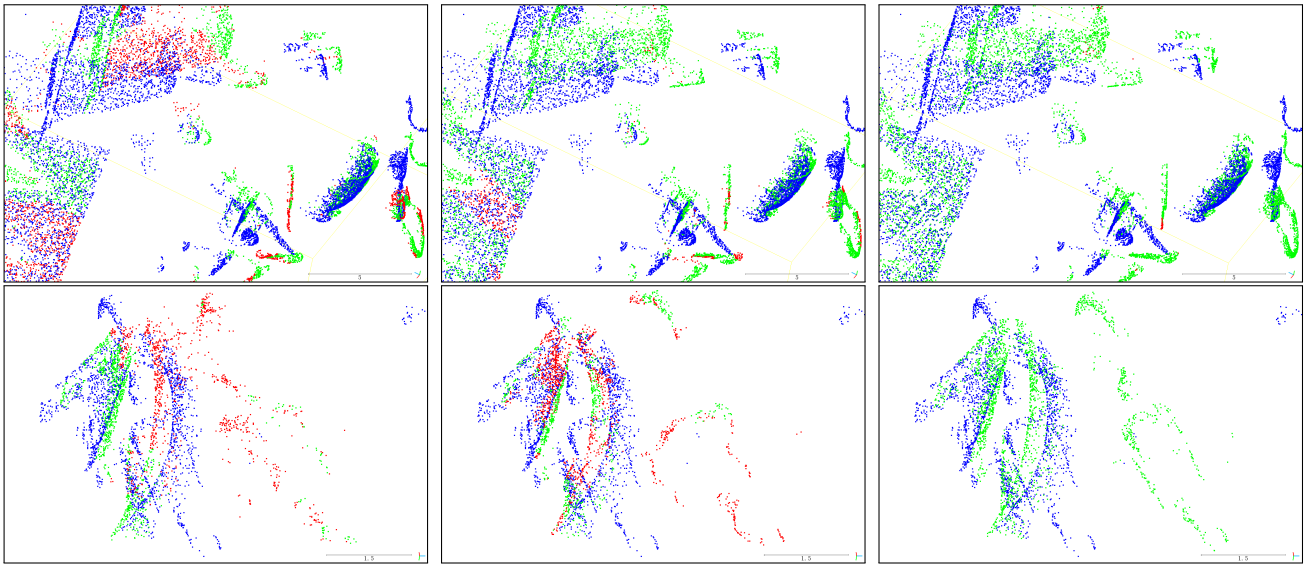
Experiments are conducted on four datasets, namely the synthetic dataset FlyThings3D (FT3D) [23] and three real-world datasets, including Stereo-KITTI [24, 25], LiDAR-KITTI [10], and SF-KITTI [8].

The synthetic dataset FT3D is derived from a large-scale collection of stereo videos. Each pair of point cloud scenes is generated from RGB stereo images in the ShapeNet [3], with random motion assigned to multiple objects within the scenes. Another stereo dataset Stereo-KITTI is a real-world dataset comprising 200 training sets and 200 testing sets. Building on the previous literature, two distinct pre-processing techniques are employed for these two datasets. A technique from [13] is utilized to remove points that do not correspond between consecutive frames, resulting in processed datasets referred to as FT3Ds and KITTIs. The second technique, proposed by [21], takes a different approach by preserving the occluded points. Instead of removing them, mask labels are used to indicate occluded points for evaluation purposes. The datasets generated using this method are referred to as FT3Do and KITTIo. The FT3Ds dataset comprises 19,640 pairs of training data and 3,824 pairs for evaluation. On the other hand, the FT3Do dataset consists of 20,008 point cloud pairs for training and 2,008 pairs for testing. Regarding the KITTIs and KITTIo datasets, they contain 142 and 150 pairs of test-only data, respectively.

However, it is noteworthy that the FT3D and Stereo-KITTI datasets, which are derived from dense and regular disparity images, differ from real-world LiDAR-scanned datasets. To demonstrate the robust generalization of our SSRFlow on LiDAR-scanned datasets, we conducted additional experiments using datasets [8, 10, 12]. Specifically, SF-KITTI [8] is a large-scale real-world scene flow dataset that is based on LiDAR-scanned data. It comprises 7,185 pairs of data. Following [8], we further divide the dataset into 6,400 pairs for training and 600 pairs for testing. In addition, the LiDAR-KITTI [10, 12], dataset includes 142 pairs of real-world scenarios captured by the Velodyne 64-beam LiDAR, specifically designed for testing purposes. This dataset is particularly valuable as it highlights the sparse and non-corresponding point characteristics that are often observed between consecutive frames of the real-world LiDAR-scanned point clouds.

11. More Results





PointPWC[40] Bi-PointFlow[4] SSRFlow

Figure 13. The blue points denote the source frame. The green points represent the result of warping the source frame using predictions. Here, the red points indicate wrong predicted warped points that have an EPE3D greater than 0.1m.