2023 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)

2023 IEEE International Conference on Multimedia and Expo Workshops (ICMEW) | 979-8-3503-1315-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICMEW59549.2023.00080

Spatial-Temporal Consistency Refinement Network for Dynamic Point Cloud Frame Interpolation

Lancao Ren
University of Electronic Science and Technology of China
Chengdu, China
lancaoren@foxmail.com

Lili Zhao
China Mobile Research Institute
China Mobile Communications Co., Ltd.
Beijing, China
zllmail@foxmail.com

Zhuoqun Sun
University of Electronic Science and Technology of China
Chengdu, China
zhuoqunsun@foxmail.com

Zhipeng Zhang
China Mobile Research Institute
China Mobile Communications Co., Ltd.
Beijing, China
zzp_zzp2002@aliyun.com

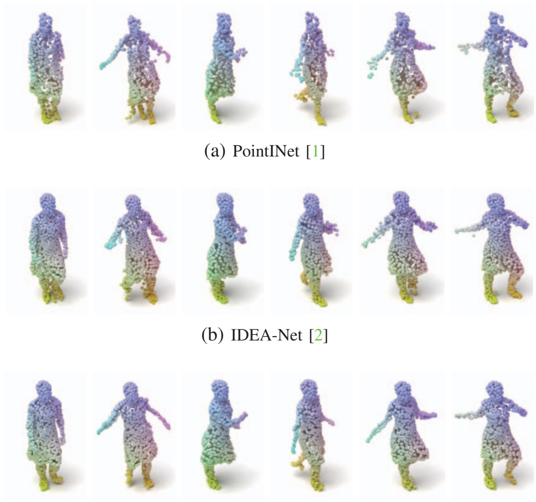
Jianwen Chen
University of Electronic Science and Technology of China
Chengdu, China
chenjianwen@uestc.edu.cn

***Abstract*—Point cloud frame interpolation aims to improve the frame rate of a point cloud sequence by synthesising intermediate frames between consecutive frames. Most of the existing works only use the scene flow or features, not fully exploring their local geometry context or temporal correlation, which results in inaccurate local structural details or motion estimation. In this paper, we organically combine scene flows and features to propose a two-stage network based on residual-learning, which can generate spatially and temporally consistent interpolated frames. At the Stage 1, we propose the spatial-temporal warping module to effectively integrate multi-scale local and global spatial features and temporal correlation into a fusion feature, and then transform it into a coarse interpolated frame. At the Stage 2, we introduce the residual-learning structure to conduct spatial-temporal consistency refinement. A temporal-aware feature aggregation module is proposed, which can facilitate the network adaptively adjusting the contributions of spatial features from input frames, and predict the point-wise offset as the compensations due to coarse estimation errors. The experimental results demonstrate our method achieves the state-of-the-art performance on most benchmarks with various interpolated modes. Code is available at https://github.com/renlancao/SR-Net.**

***Index Terms*—point cloud, frame interpolation, spatial-temporal consistency, residual learning**

I. Introduction

With rapid developments in 3D acquisition equipment, point cloud has become one of the most widely-used 3D digital representation for real-world persons, objects or scenes, which can be encountered in many applications, such as AR/VR/XR, urban digital twins, Metaverse, etc. However, limited by sensor performance and high cost of acquisition, the frame rate of captured dynamic point clouds (i.e., a sequence, formed by consecutive frames along the time) is typically low (about 25 Hz), which leads to poor temporal consistency. It is known that higher-frame-rate sequences could bring more immersive visual experience for users. Therefore, point cloud frame interpolation, which can increase the frame rate of a point cloud sequence by generating the intermediate frames between consecutive frames, is needed.


(a) PointINet [1]

(b) IDEA-Net [2]

(c) Ours

Fig. 1. Visualization of interpolated results on the *Swing*.

Recently, several deep learning-based frame interpolation methods (e.g, [1]–[4]) for dynamic point clouds have been proposed, which can be divided into two categories: the 2D space-based methods (e.g., [3], [4]) and the 3D space-based methods (e.g., [1], [2]). The 2D space-based methods usually exploited the 2D range image (RI) representation of LiDAR point clouds and used the video frame interpolation algorithms. However, the RI representation is only applicable to *sparse* point clouds acquired by LiDAR sensors, which often used in

*Zhuoqun Sun, Lili Zhao and Lancao Ren contribute equally. Work completed during Lancao Ren and Zhuoqun Sun's internship at China Mobile Research Institute.

979-8-3503-1315-4/23/$31.00 ©2023 IEEE
DOI 10.1109/ICMEW59549.2023.00080

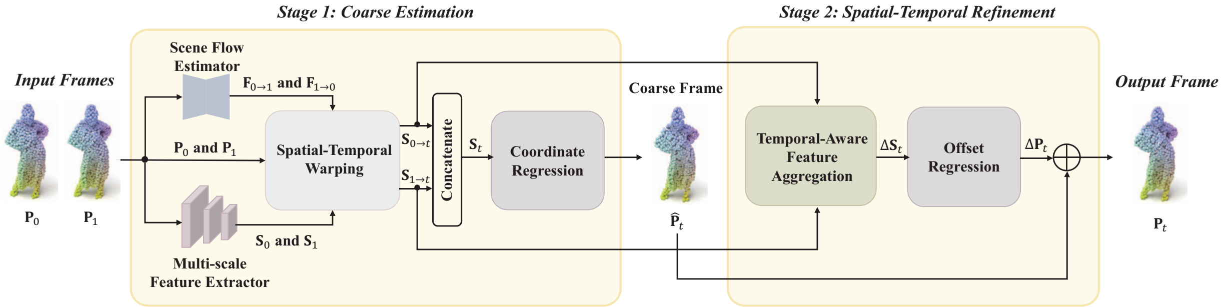


Fig. 2. The overall pipeline of our proposed network. $\oplus$ denotes the element-wise sum operation.

autonomous driving and mobile robots, etc. In contrast, *dense* point clouds that have larger point density and more geometric details, are more for human vision and used in multimedia applications. In this paper, we will put our eyes on *dense* point cloud frame interpolation.

Instead of using 2D RIs, several works ([1], [2], [5]) explored frame interpolation directly in 3D space. For example, PointINet [1] used the existing scene flow estimation network (i.e., FlowNet3D [6]) with warping operation to get a coarse frame, and then proposed a point fusion module to refine it into the final interpolated frame. However, it synthesizes the coarse frame only based on scene flow and input frames, which is under the assumption that the motion is linear. As a result, it fails to reconstruct local geometric details, especially for large deformation, as shown in Fig. 1 (a). Instead of using scene flows, IDEA-Net [2] introduced high-dimensional features, which proposed a temporal consistency module to estimate 3D motion based on features, and then generated the interpolated frames in a coarse-to-fine manner. However, IDEA-Net [2] set the number of interpolated frames as the input and output dimension of convolution layers in the temporal consistency module. This means that a well-trained IDEA-Net can not change the number of interpolated frames, unless training it from scratch, which limits its application to conduct arbitrary time-interval frame interpolation. Moreover, in IDEA-Net [2], two candidate interpolated frames are predicted in dual branches separately, based on the one of input two frames (i.e., reference frames), respectively. More notably, the final interpolated frame is chosen by comparing whose reference frame has the smaller time interval difference from the target interpolated frame. Obviously, the spatial-temporal consistency of two input frames is not fully exploited simultaneously, leading to its poor performance in local structural details reconstruction and long-term motion consistency, as shown in Fig. 1 (b).

To deal with these limitations, we propose a two-stage network based on residual learning, which organically combines scene flows and features, while exploiting the spatial-temporal consistency of two input frames simultaneously. At the Stage 1, we propose a spatial-temporal warping module to integrate multi-scale spatial features and 3D flows with input frames into one coarse frame, which facilitates better geometry details reconstruction and temporal modeling. At the Stage 2, we introduce residual learning to refine the coarse interpolated frame and propose a temporal-aware feature aggregation module. This module makes the network can dynamically adjust the contributions of the spatial features from input two frames, and then predicts the point-wise offset to refine the coarse frame obtained in the Stage 1. The spatial-temporal consistency is implicitly enforced by two-stage operations. The main contributions of this paper are as follows:

- We propose a network to improve the frame rate of dynamic point clouds. Our model can conduct spatially-temporally consistent point cloud frame interpolation with varying numbers of interpolated frames.
- We leverage spatial-temporal consistency by the spatial-temporal warping module and temporal-aware feature aggregation module in a coarse-to-fine manner. The ablation study proves their efficiency.
- Quantitative and qualitative experiments have demonstrated that our proposed method is comparable or outperforms the existing state-of-the-art methods.

II. Method

A. Problem Setting

Given two consecutive input point cloud $\mathbf{P}_0 \in \mathbb{R}^3$ and $\mathbf{P}_1 \in \mathbb{R}^3$, our goal is to generate the intermediate frame $\mathbf{P}_t \in \mathbb{R}^3$ at time $t \in (0, 1)$ as

$$\mathbf{P}_t = f\left(\mathbf{P}_0, \mathbf{P}_1, t\right), \qquad (1)$$

where f is the desired frame interpolation network. Note that the number of interpolated frames is not fixed.

B. Overview

Fig. 2 shows the overall scheme of our proposed method. It consists of two stages: 1) coarse estimation using scene flow estimator, multi-scale feature extractor and spatial-temporal warping, and 2) spatial-temporal refinement via temporal-aware feature aggregation and residual learning.

Given two input frames $\mathbf{P}_0$ and $\mathbf{P}_1$, the bi-directional scene flows (denoted as $\mathbf{F}_{0\to1}$ and $\mathbf{F}_{1\to0}$), are firstly estimated by the scene flow estimator [7]. Next, by our proposed multi-scale feature extractor, we extract the global and local spatial features of input frames, $\mathbf{S}_0$ and $\mathbf{S}_1$. With the scene flows $\mathbf{F}_{0\to1}$ and $\mathbf{F}_{1\to0}$, we warp the input frames and their spatial features via a designed spatial-temporal warping module. As a result, the spatial features of the frame at time t can be obtained, which refer to one of two input frames, respectively, denoted as $\mathbf{S}_{0\to t}$ and $\mathbf{S}_{1\to t}$. Next, $\mathbf{S}_{0\to t}$ and $\mathbf{S}_{1\to t}$ are concatenated into a spatial-temporal feature $\mathbf{S}_t$. Then, we transform $\mathbf{S}_t$ into a coarse frame $\hat{\mathbf{P}}_t$ via coordinate regression.

At the Stage 2, we introduce the residual learning structure to refine $\hat{\mathbf{P}}_t$, while further exploiting the spatial-temporal consistency. First, aggregate spatial features $\mathbf{S}_{0\to t}$ and $\mathbf{S}_{1\to t}$ into a feature offset $\Delta\mathbf{S}_t$ via a temporal-aware feature aggregation module. Then, derive the per-point offset $\Delta\mathbf{P}_t$ by offset regression. Finally, a refined interpolated frame $\mathbf{P}_t$ can be generated by combining $\Delta\mathbf{P}_t$ and the coarse frame $\hat{\mathbf{P}}_t$. In what follows, the main modules will be respectively described in detail.

C. Stage 1: Coarse Estimation

Multi-Scale Feature Extractor. To get the spatial feature of each input frame, we extend the DGCNN [8] into a multi-scale structure. The principle behind this design is that the local and global semantics could be jointly learned by the multi-scale extraction strategy [9]–[12]. Specially, we first use furthest point sample (FPS) to divide the input point cloud into three sets ($\mathbf{P}_{l_0}, \mathbf{P}_{l_1}, \mathbf{P}_{l_2} \in \mathbb{R}^3$) at various scales. For the scale l_0, the input point cloud $\mathbf{P}_{l_0}$ is fed into the first EdgeConv [8] to get the aggregated output feature $\mathbf{f}_{l_0}$. Then, the output feature $\mathbf{f}_{l_0}$ and $\mathbf{P}_{l_1}$ are as the input of the second and third EdgeConv [8] to get the features $\mathbf{f}_{l_1}$ and $\mathbf{f}_{l_2}$ in sequence. Finally, $\mathbf{f}_{l_2}$ and $\mathbf{P}_{l_2}$ are fed into the last EdgeConv [8] to derive the final features $\mathbf{S}_0$ and $\mathbf{S}_1$. More implementation details are provided in the *supplementary materials*.

Spatial-Temporal Warping. This module contains two steps: (i) warp each input frames ($\mathbf{P}_0$, $\mathbf{P}_1$) and its scene flows ($\mathbf{F}_{0\to1}$, $\mathbf{F}_{1\to0}$) estimated by [7], and then get two warped frames ($\mathbf{P}_{1\to t}$, $\mathbf{P}_{0\to t}$). (ii) Based on the point-to-point relationship between each input frame and its warped frame ($\mathbf{P}_0$ and $\mathbf{P}_{0\to t}$, $\mathbf{P}_1$ and $\mathbf{P}_{1\to t}$), project $\mathbf{S}_0$ and $\mathbf{S}_1$ into the feature of the interpolated frame, $\mathbf{S}_{0\to t}$ and $\mathbf{S}_{1\to t}$, respectively.

Specifically, for each point p in the warped frame $\mathbf{P}_{0\to t}$, we first find its neighbor points in the corresponding input frame $\mathbf{P}_0$, and then calculate the distances between them. Next, the warped spatial features $\mathbf{S}_{0\to t}$ can be estimated as

$$\mathbf{S}_{0\to t}(p) = (\sum_{k\in\mathcal{N}(p;\mathbf{P}_0)} \mathbf{S}_0^{(k)} \cdot w_k)/(\sum_{k\in\mathcal{N}(p;\mathbf{P}_0)} w_k), \tag{2}$$

where $\mathbf{S}_0^{(k)}$ refers to the feature of the point with index k in $\mathbf{P}_0$, and $\mathcal{N}(p;\mathbf{P}_0)$ denotes the index set of the neighbor points. w_k can be calculated based on distances between p and its neighbor point $\mathbf{P}_0^{(k)}$ as

$$w_k = \frac{1}{\left\| p - \mathbf{P}_0^{(k)} \right\|_2}\ ,\ k \in \mathcal{N}(p;\mathbf{P}_0). \tag{3}$$

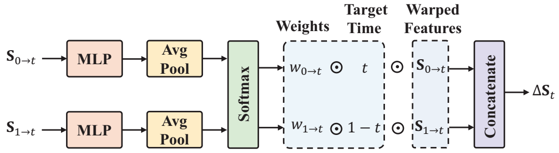


Fig. 3. Illustration of the proposed temporal-aware feature aggregation module. $\odot$ denotes the multiplication operation.

Note that $\mathbf{S}_{1\to t}$ can be obtained in the same way based on $\mathbf{P}_1$ and $\mathbf{P}_{1\to t}$. Then, $\mathbf{S}_{0\to t}$ and $\mathbf{S}_{1\to t}$ are concatenated into a spatial-temporal feature $\mathbf{S}_t$.

Coordinate Regression. We reconstruct a coarse frame $\hat{\mathbf{P}}_t$ from $\mathbf{S}_t$ (with the size of $N \times C$), by using regression via multi-layer perceptions (MLPs).

D. Stage 2: Spatial-Temporal Refinement

Temporal-aware Feature Aggregation. To further refine the coarse frame with spatial-temporal consistency, we introduce the residual-learning structure and propose a temporal-aware feature aggregation module. As depicted in Fig. 3, the warped spatial features $\mathbf{S}_{0\to t}$ and $\mathbf{S}_{1\to t}$ are firstly fed into a shared MLP and an average pooling layer, respectively. Then, we calculate two adaptive weights ($w_{1\to t}$ and $w_{0\to t}$) by using the softmax function, which makes the network dynamically adjust the contributions of the spatial features based on the interpolated time t. Finally, the aggregated feature $\Delta\mathbf{S}_t$ can be obtained by concatenate operation.

Offset Regression. After obtained the aggregated feature $\Delta\mathbf{S}_t$, we use the residual learning structure to predict the point-wise coordinate offsets $\Delta\mathbf{P}_t$. Same as the coordinate regression, offset regression is realized by a set of MLPs. Then, the refined interpolated frame can be derived by

$$\mathbf{P}_t = \hat{\mathbf{P}}_t + \Delta\mathbf{P}_t. \tag{4}$$

E. Loss function

To evaluate the difference between two point clouds, Chamfer distance (CD) is used as the loss function in our work. Given the ground-truth point cloud frame $\mathbf{G}_t \in \mathbb{R}^3$ and the interpolated one $\mathbf{P}_t \in \mathbb{R}^3$ at time $t \in [0,1]$, CD can be formulated as

$$\mathcal{L}_{CD} = \frac{1}{N_{\mathbf{P}_t}} \sum_{x\in\mathbf{P}_t} \min_{y\in\mathbf{G}_t} \|x-y\|_2^2 + \frac{1}{N_{\mathbf{G}_t}} \sum_{x\in\ \mathbf{G}_t} \min_{y\in\mathbf{P}_t} \|y-x\|_2^2\,, \tag{5}$$

where $\|\cdot\|_2$ represents the ℓ_2 norm.

III. Experiments

A. Experimental Settings

Datasets. We select fifteen sequences from MITAMA dataset [13] and 8iVSLF dataset [14], which contain dynamic 3D meshes and real-scanned point clouds, respectively. Following IDEA-Net [2], all the sequences are uniformly downsampled to 1024 points. For MITAMA [13], we used eight

TABLE I
QUANTITATIVE ($\times 10^{-3}$) COMPARISONS ON THE MITAMA [13] DATASET AND 8IVSLF DATASET [14]. PREDICTING x INTERMEDIATE FRAME(S) BASED ON THE CONSECUTIVE TWO FRAMES (DENOTED AS $2 \rightarrow x$). THE SYMBOL '-' MEANS THE RESULTS ARE UNAVAILABLE.

Interpolation Mode	Method	Swing		Squat2		Longdress		Loot		Thaidancer		Average	
		CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD
$2 \rightarrow 1$	IDEA-Net [2]	-	-	-	-	-	-	-	-	-	-	-	-
	PointINet [1]	1.14	15.49	1.25	20.50	1.09	13.00	1.06	14.95	1.35	21.93	1.18	17.17
	Ours	**0.51**	**1.46**	**0.42**	**0.94**	**0.82**	**5.44**	**0.80**	**7.04**	**0.91**	**6.82**	**0.69**	**4.34**
$2 \rightarrow 2$	IDEA-Net [2]	-	-	-	-	-	-	-	-	-	-	-	-
	PointINet [1]	1.55	19.89	1.45	25.32	1.22	16.65	1.21	18.94	1.48	26.57	1.38	21.47
	Ours	**0.93**	**3.84**	**0.63**	**1.56**	**0.92**	**6.16**	**0.88**	**7.50**	**0.99**	**7.20**	**0.87**	**5.25**
$2 \rightarrow 3$	IDEA-Net [2]	1.24	7.07	**0.24**	**0.45**	**0.87**	**5.95**	**0.83**	8.21	1.16	8.40	**0.87**	6.00
	PointINet [1]	2.06	22.00	1.55	26.78	1.28	18.11	1.25	20.49	1.52	27.58	1.53	22.99
	Ours	**1.01**	**5.03**	0.48	1.05	1.02	6.64	0.95	**8.03**	**1.04**	**7.81**	0.90	**5.71**
$2 \rightarrow 4$	IDEA-Net [2]	-	-	-	-	-	-	-	-	-	-	-	-
	PointINet [1]	2.75	24.52	1.70	28.57	1.34	18.98	1.31	21.61	1.56	28.72	1.73	24.48
	Ours	**1.63**	**8.17**	**0.90**	**2.86**	**1.18**	**8.58**	**1.07**	**10.32**	**1.17**	**9.94**	**1.19**	**7.97**
$2 \rightarrow 5$	IDEA-Net [2]	-	-	-	-	-	-	-	-	-	-	-	-
	PointINet [1]	3.44	26.30	1.88	30.83	1.39	19.03	1.34	21.90	1.59	29.77	1.93	25.56
	Ours	**2.48**	**14.68**	**1.38**	**6.87**	**1.36**	**10.43**	**1.21**	**12.07**	**1.37**	**11.86**	**1.56**	**11.18**

TABLE II
QUANTITATIVE ($\times 10^{-3}$) COMPARISONS AT DIFFERENT INTERPOLATED TIME t FOR $2 \rightarrow 3$.

t	Method	Swing		Longdress		Loot		Thaidancer	
		CD	EMD	CD	EMD	CD	EMD	CD	EMD
0.25	IDEA-Net [2]	1.09	5.11	0.89	5.84	0.85	8.21	1.23	9.41
	PointINet [1]	1.89	23.76	1.32	20.78	1.30	22.19	1.58	30.44
	Ours	**0.88**	**4.11**	**0.82**	**5.24**	**0.81**	**6.73**	**0.90**	**6.17**
0.5	IDEA-Net [2]	1.42	8.94	**0.93**	6.67	**0.86**	8.55	1.18	8.03
	PointINet [1]	2.37	20.32	1.21	13.29	1.14	15.92	1.41	21.35
	Ours	**1.19**	**6.48**	1.02	**6.48**	0.94	**7.78**	**1.03**	**7.50**

sequences for training and two for testing. For 8iVSLF [14], we used two sequences for fine-tuning and three for testing. More details are provided in the *supplementary materials*.

Training Details. We first fine-tune the scene flow estimation model Bi-PointFlowNet [7] on our train dataset for 1000 epochs, with the initial learning rate of 0.001 and the minimum learning rate of 0.00001. Then, the model's parameters are freezed. Then, we train the frame interpolation model for another 900 epochs, where the Adam optimizer is used with the initial learning rate of 0.001, and the minimum learning rate of 0.00004. We set the batch size to 14 and use PyTorch to implement our model on one GeForce RTX 3090 GPU.

Evaluation Metrics. To evaluate the performance, Chamfer distance (CD) and Earth Mover's Distance (EMD) are used. CD measures the similarity between two point clouds by calculating the average distance of each point in one set to their nearest neighbor(s) in the other set, while EMD measures the similarity by finding a point-to-point bijection with the minimum distance between point clouds. CD has been described in Eq. (5), and EMD can be calculated as

$$\mathcal{L}_{EMD} = \min_{\theta:\mathbf{P}_t \rightarrow \mathbf{G}_t} \frac{1}{N} \sum\nolimits_{x \in \mathbf{P}_t} \| \, x - \theta(x) \|_2, \qquad (6)$$

where $\mathbf{G}_t$ and $\mathbf{P}_t$ represent two point clouds, and N is the point number, while $\theta : \mathbf{P}_t \rightarrow \mathbf{G}_t$ is a bijection.

B. Comparisons with State-of-the-Art Methods

To prove the effectiveness of our proposed algorithm, we compare our method with the existing competitive works, including PointINet [1] and IDEA-Net [2].

Quantitative Comparison. Table I shows the CD and EMD values for testing on MITAMA dataset [13] and 8iVSLF dataset [14]. Note that for a well-trained IDEA-Net [2], the number of interpolated frames is fixed. The training code of IDEA-Net [2] is not publicly available, which means it can not conduct various-length frame interpolation such as $2 \rightarrow 4$ or $2 \rightarrow 5$. Therefore, we denote these cases as '-'. It can be observed that our proposed method outperforms PointINet [1] for all the interpolation modes on all test datasets. Note that in IDEA-Net [2], the final interpolated frame is chosen from two candidates derived by two branches by comparing whose reference frame has the smaller time interval from the target frame. When the time interval is small (e.g., $2 \rightarrow 3$), the sequences are often with smaller motion. Therefore, this strategy may be effective. However, as shown in Table I, our method is more competitive in other cases, especially for the sequences with larger motion. Besides, Table II provides quantitative comparisons on frame interpolation at any time (e.g., $t = 0.25$ and $t = 0.5$) for $2 \rightarrow 3$. From Table II, we observed that our method delivers the best performance on most of the test sequences at a specified time.

Qualitative Comparisons. Fig. 4 shows the qualitative results of different methods on MITAMA dataset [13] and 8iVSLF dataset [14]. As shown in the zoom-in image patches, it can

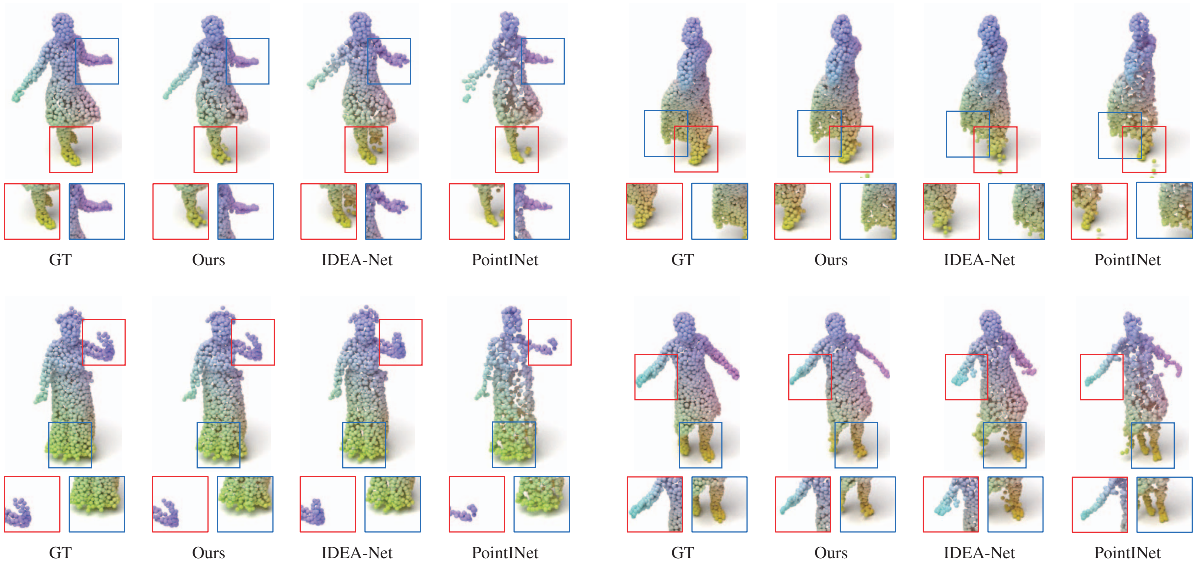


Fig. 4. Qualitative comparisons with PointINet [1] and IDEA-Net [2] on the MITAMA [13] dataset and 8iVSLF [14] dataset. The interpolated mode is 2 → 3. The frames shown are from Sequence *Swing* (top-left), *Longdress* (top-right), *Thaidancer* (bottom-left), *Swing* (bottom-right).

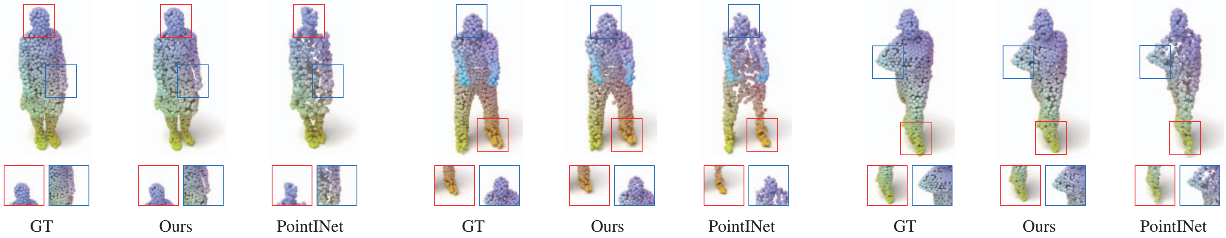


Fig. 5. Qualitative comparisons with PointINet [1] on the MITAMA [13] dataset and 8iVSLF [14] dataset, while the interpolated mode is 2 → 4. The frames shown are from Sequence *Swing*, *Loot* and *Longdress*, from left to right.

be obviously observed that our method produces finer and more reliable geometric details even when the structure is complicated and motion is large, such as hands or feet of persons, skirts. Besides, we also give visual comparisons with PointINet [1] when the interpolation mode is 2 → 4. It can be clearly seen that our method still has a great advantage in local structure reconstruction, even if there is a large time interval. More visual comparison results with PointINet [1] on various-length frame interpolation (i.e. 2 → 1, 2 → 2, 2 → 4 and 2 → 5) are provided in our *supplementary material*.

Both the quantitative and qualitative results demonstrate that the proposed method achieves superior or comparable point cloud frame interpolation results than existing methods.

C. Ablation Study

We verify the effect of different components of the proposed model: (i) replacing the multi-scale extractor with the general DGCNN [8]; (ii) removing the spatial-temporal warping module, where only the features of input frames are concatenated and fed into coordinate regression; (iii) removing the interpolated time t in the temporal-aware feature aggregation module; (iv) removing the Stage 2 (spatial-temporal refinement) and directly using the coarse frame as output. Each model is trained with the same training strategy and tested on the same dataset. Table III summarizes the results of each case, compared to our full network (the bottom row). It can be seen that our full network delivers the best performance.

IV. Conclusion

In this paper, we propose a two-stage point cloud frame interpolation network via residual-learning, which can generate spatially and temporally consistent interpolated frames to increase the frame rate of dynamic point clouds. The success of our approach is due to the fact that we combine the scene flow and features to jointly exploit the spatial-temporal correlation

TABLE III
IMPACTS OF THE MULTI-SCALE FEATURE EXTRACTOR, SPATIAL-TEMPORAL WARPING MODULE, TEMPORAL-AWARE FEATURE AGGREGATION MODULE AND OFFSET REFINEMENT IN OUR MODEL.

Model	CD ($\times10^{-3}$)	EMD ($\times10^{-3}$)
General DGCNN as the feature extractor	2.13	20.93
w/o spatial-temporal warping	0.94	5.72
w/o t in temporal-aware feature aggregation	0.98	5.76
w/o offset	1.09	6.80
The full network	**0.90**	**5.71**

of two input frames. Extensive experiments have demonstrated that the proposed method can consistently deliver state-of-the-art performance on most cases in terms of objective evaluation and subjective assessment, while the number of interpolated frames is unrestricted. All these have clearly shown that our proposed method can be widely used into a set of immersive applications.

REFERENCES

[1] Fan Lu, Guang Chen, Sanqing Qu, Zhijun Li, Yinlong Liu, and Alois Knoll, "PointINet: Point cloud frame interpolation network," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021, pp. 2251–2259.

[2] Yiming Zeng, Yue Qian, Qijian Zhang, Junhui Hou, Yixuan Yuan, and Ying He, "IDEA-Net: Dynamic 3D point cloud interpolation via deep embedding alignment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6328–6337.

[3] Lili Zhao, Xuhu Lin, Wenyi Wang, Kai-Kuang Ma, and Jianwen Chen, "RangeINet: Fast LiDAR point cloud temporal interpolation," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 2584–2588.

[4] Lili Zhao, Zezhi Zhu, Xuhu Lin, Xuezhou Guo, Qian Yin, Wenyi Wang, and Jianwen Chen, "RAI-Net: Range-adaptive LiDAR point cloud frame interpolation network," in *Proceedings of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2021, pp. 1–6.

[5] Anique Akhtar, Zhu Li, Geert Van der Auwera, and Jianle Chen, "Dynamic point cloud interpolation," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 2574–2578.

[6] Xingyu Liu, Charles R. Qi, and Leonidas J. Guibas, "FlowNet3D: Learning scene flow in 3D point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 529–537.

[7] Wencan Cheng and Jong Hwan Ko, "Bi-PointFlowNet: Bidirectional learning for point cloud based scene flow estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022, pp. 108–124.

[8] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 5, pp. 1–12, 2019.

[9] Ruth Wijma, Shaodi You, and Yu Li, "Multi-level adaptive separable convolution for large-motion video frame interpolation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2021, pp. 1127–1135.

[10] Wang Yifan, Shihao Wu, Hui Huang, Daniel Cohen-Or, and Olga Sorkine-Hornung, "Patch-based progressive 3D point set upsampling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5958–5967.

[11] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou, "PoinTr: Diverse point cloud completion with geometry-aware transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 12478–12487.

[12] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 5099–5108.

[13] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popovic, "Articulated mesh animation from multi-view silhouettes," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3, pp. 1–9, 2008.

[14] Maja Krivokuća, Philip A. Chou, and Patrick Savill, "8i voxelized surface light field (8iVSLF) dataset," *ISO/IEC JTC1/SC29 WG11 (MPEG) input document m42914*, vol. 7, pp. 8, 2017.