Sequential Point Clouds: A Survey Supplementary

Haiyan Wang and Yingli Tian, *Fellow, IEEE,*

Abstract—Point clouds have garnered increasing research attention and found numerous practical applications. However, many of these applications, such as autonomous driving and robotic manipulation, rely on sequential point clouds, essentially adding a temporal dimension to the data (i.e., four dimensions) because the information of the static point cloud data could provide is still limited. Recent research efforts have been directed towards enhancing the understanding and utilization of sequential point clouds. This paper offers a comprehensive review of deep learning methods applied to sequential point cloud research, encompassing dynamic flow estimation, object detection & tracking, point cloud segmentation, and point cloud forecasting. This paper further summarizes and compares the quantitative results of the reviewed methods over the public benchmark datasets. Ultimately, the paper concludes by addressing the challenges in current sequential point cloud research and pointing towards promising avenues for future research.

Index Terms—4D sequential point cloud; Deep learning; Flow estimation; Object detection & tracking; Point cloud segmentation; Point cloud forecasting.

1 Common Deep Network Architectures

In this section, we briefly summarize the common deep networks for general feature learning of high dimensional data. There are mainly two streams methods. One is applying the convolution neural networks to directly learn the spatio-temporal features (Sec. 1.1). Another stream is adopting the recurrent networks and recurrently comprehending the hidden states (Sec. 1.2).

1.1 Convolution Neural Network

According to the representation of the input data, these methods can be categorized into grid-based and point-based architectures.

1.1.1 Grid-based Architectures

These methods transfer point clouds into regular representations such as voxel or point tube, which could further support the common convolution layers to extract features. Figure 1 shows a standard grid-based network for feature learning.

4D MinkNet [3] was the first one to exploit the common deep learning network on high-dimensional data. It adopted the idea from Sparse Tensor [9] and proposed a generalized sparse convolution to operate SPL. The proposed convolution layer can be blended with various deep networks and well generalized to different tasks. To deal with the computational problem when generalizing the convolution to high dimensional spaces, the authors designed a novel kernel that is not hyper cubic and thus diminishes the memory cost. Moreover, the high-dimensional conditional random fields were introduced to enforce the consistency between the space and time domains. Incorporating all of these designs, MinkNet was established to provide a general deep network to handle sequential point clouds.

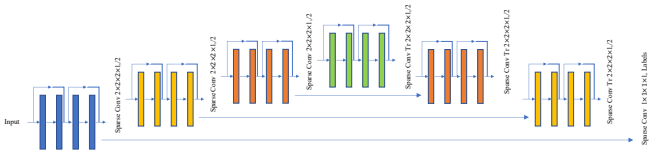

Fig. 1: The illustration of a Grid-based Architecture. The figure is reproduced based on [3].

PSTNet [6] was another grid-based method which performs spatio-temporal convolution on the sequential point clouds. They decoupled the spatial and temporal information from the input raw point clouds which is shown to be more reasonable and effective. Unlike the above-mentioned method MinkNet [3], which are based on sparse convolution, PSTNet developed a Point tube structure to manage the input data and conduct the proposed convolution. The point tube incorporated the spatial and temporal kernels separately to capture the spatio-temporal local structure information. To tackle both the sequence-level and point-level classification tasks such as semantic segmentation, the authors introduced the PST convolutions and transposed convolutions to construct the PSTNet hierarchically.

1.1.2 Point-based Architectures

These methods assemble the network based on a set of MLP (Multiple Layer Perception) layers and aggregate feature information from neighborhood regions along with both spatial and temporal domains. We show a point-based architecture in Figure 2.

MeteorNet [11] was a seminal work that explores deep learning on the SPL data with a direct point-based method. Treating PointNet++ [15] as an elementary unit, it proposed an architecture to digest the input point cloud sequences by early/later fusion methods, which provided a common solution to learn sequential point cloud features. Moreover, another core contribution was that they fused

- *Haiyan Wang is with the Department of Electrical Engineering, The City College of New York, New York, NY, 10031. E-mail: hwang3@ccny.cuny.edu*
- *Yingli Tian (Corresponding author) is with the Department of Electrical Engineering, The City College, and the Department of Computer Science, the Graduate Center, the City University of New York, New York, NY, 10031. E-mail: ytian@ccny.cuny.edu*

This material is based upon work supported by the National Science Foundation under award number IIS-2041307.

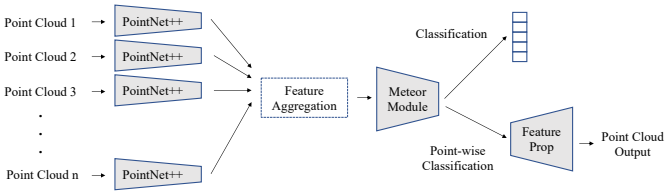

Fig. 2: The illustration of a point-based architecture build upon [11].

the temporal information by explicitly grouping the meaningful neighbor regions. Two grouping methods were proposed to solve the problem, direct grouping and chain-flow grouping. The direct grouping-based method increased the radius monotonically with the time increases to search the nearest neighbor region. The chain-flow-based method predicted the flow using the offline network Flownet3D [10] to better find the correspondence between frames. The scene flow could help to find a better search radius to confirm the nearest neighbor region.

1.2 Recurrent Neural Network

Besides the convolution networks, the recurrent neural network is another intuitive method to process SPL data. PointRNN [5] proposed a Point Recurrent Neural Network (PointRNN) to learn the representation of moving point clouds. PointRNN extended the idea from the 2D RNN to the 3D/4D RNN. Meanwhile, due to the problem of representing the point cloud into a single state vector, PointRNN separated the position features and auxiliary features to sever as the state vectors for updating. Another problem when applying RNN to point clouds is that simple concatenation of data over the temporal series cannot be conducted due to the point clouds being unordered. Instead, PointRNN tackled this problem by adopting a correlation layer between the previous state vector and current input data. It would search the nearest neighbor points linked to the query point and concatenate them separately. The final pooling operation would extract a single vector from the previous representations. The authors affirmed the effectiveness of LSTM for the moving point cloud prediction task.

1.3 Discussion

All of the above-mentioned deep learning methods investigated general pipelines to conduct feature learning from SPL data. Here we briefly summarize the characters of different network architectures:

- Convolution Neural Networks exploit operations over the entire spatial and temporal domain. The extracted spatial and temporal features have a more mutual impact. Thus these networks focus more on the feature consistency along temporal sequence. Some high-level tasks which require a better semantic understanding such as detection and segmentation will benefit more from feature learning of Convolution Neural Network.
- Essentially, Recurrent Neural Networks emphasize more on the long-range dependency along the time dimension. The temporal relation between distinct frames is explicitly represented by their recurrent design. Thus, those long-range sequence tasks are more appropriate with Recurrent Neural Network such as action recognition or object tracking.

2 DATASETS

We summarize the datasets commonly used in SPL analysis in Table 1.

ATG4D Dataset [14] consists of 5,000 sequences of training data including total 1.2 million Lidar sweeps, while the testing set contains 500 sequences and 5,969 sweeps. The dataset is captured by the Velodyne 64E LiDAR and is mainly used for motion forecasting tasks.

Flythings3D Dataset [12] is currently the largest synthetic dataset for scene flow estimation which contains about 22,000 stereo frames with spatial size of 960x540 pixels. These images are rendered from randomized synthetic sequences along with moving objects from ShapeNet [19]. The ground truth annotations include segmentation maps, disparity maps, disparity changes, and optical flow maps. Thus, the point cloud sequences can be reconstructed from the disparity maps and the corresponding groundtruth scene flow annotations can be obtained through back-projecting the 2D optical flow maps and disparity changes to the 3D space.

KITTI Dataset [8] is popular dataset and widely used for object detection and tracking tasks. It consists a total of 50 sequences with the split of 16 training sequences, 5 offline testing sequences and 29 online testing sequences. Specifically, sequence 0 to 15 are adopted for training and sequence 16 to 20 for offline testing.

KITTI Raw Dataset [7] is adopted for the sequential foresting task which is a superset for other KITTI versions such KITTI Scene flow, detection, etc. The raw KITTI dataset contains a total of 151 sequences of Lidar data, which is divided into 60, 46, and 45 for train, val, and test respectively.

KITTI Scene Flow 2015 Dataset [13] is a real dataset that is proposed/designed for autonomous driving along with the deep learning tasks such as flow estimation, SLAM (Simultaneous Localization and Mapping), semantic & instance segmentation, depth prediction & completion, and object detection & tracking. The dataset has been collected around the city of Karlsruhe, Germany using RGBD cameras and a Velodyne 64 LIDAR scanner. Based on KITTI dataset [8], Menze et al. [13] took advantage of raw data and augmented it with detailed 3D CAD models, leading to a KITTI scene flow estimation benchmark with annotated groundtruth. There are a total of 200 training and 200 test scenes and previous research [4], [10], [18] removed the useless points belong to ground to better focus on the scene flow information. Since there is no groundtruth annotation in the testing dataset, researchers often choose 150 out of 200 training scenes as their training set and the rest 50 as their testing set.

NuScenes Dataset [2] is a large-scale autonomous driving dataset collected by Motional, with the purpose of aiding computer vision research and improving the safety of self-driving. The whole dataset consists of 1000 outdoor scenes where 850 scenes are used for training and validation, and the rest 150 scenes are for testing. There are a total of 1.4M object bounding boxes in 40k keyframes which are 7 times more object annotations than KITTI dataset [8]. The segmentation annotation covers 32 semantic categories resulting 1.4 billion annotated points across 40,000 pointclouds.

SemanticKitti [1] is built upon KITTI [8] Odometry dataset, SemanticKitti is a large-scale dataset containing semantic annotations for sequential point clouds. The Lidar point clouds are scanned at a rate of 10 Hz which help to better understand both semantic and temporal information. The whole dataset consists of 22 point clouds sequences and 43,551 point cloud frames. Specifically, they are divided with 19,130 frames (sequence 0 to 10) for training, 4,071

TABLE 1: Summary of the commonly used SPL datasets.

Datasets	Size	Annotation	Train&Val/Test	Synthetic
ATG4D [14]	5,500 sequences, 1.2M frames	Motion trajectory	5,000/500 (Frame)	×
Flythings3D [12]	22,000 pair frames	Point-wise scene flow	20,000/2,000 (Frame)	✓
KITTI [8]	21 sequences, 7,987 frames	Bounding box; Object ID	16/5 (Sequence)	×
KITTI Raw [7]	151 sequences	Raw data (Point clouds)	106/45 (Sequence)	×
KITTI Scene Flow [13]	150 pair frames	Point-wise scene flow	100/50 (Pair)	×
NuScenes Dataset [2]	1,000 scenes, 0.3M frames	Bounding box; Motion trajectory	850/150 (Scene)	×
SemanticKitti [1]	22 sequences, 43,551 frames	Point-wise class label; Object ID	19,130/4,071 (Frame)	×
Synthia 4D [16]	6 sequences, 22,589 frames	Point-wise class label	20,703/1,886 (Frame)	✓
Waymo Open Dataset [17]	1,150 sequences, 20M frames	Bounding box; Scene flow	1,000/150 (Sequence)	×

TABLE 2: Comparison of Point Cloud and Depth Image

	Point Cloud	**Depth Image**
Pros		
1. Data Representation	Points can carry diverse attributes, like color and intensity.	Structured grid format simplifies many processing tasks.
2. Flexibility	Can represent sparse or dense areas with equal ease.	Uniform resolution across the image.
3. Use Cases	Versatile for diverse applications from 3D reconstruction to Lidar processing.	Easily paired with traditional color cameras for RGB-D data.
Cons		
1. Memory Consumption	Can consume significant memory for dense data.	Fixed memory based on image resolution.
2. Processing	Requires specialized algorithms and data structures due to unordered data.	Limited to depth information unless paired with color.
3. Resolution	Density and accuracy can vary across different sections.	Cannot represent sparse data as effectively as point clouds.

frames (sequence 8) for validating and 20,351 frames (sequence 11 to 21) for testing. They provide challenges for both 3D and 4D semantic segmentation. There are a total of 25 object classes and the 3D semantic segmentation task only evaluates the performance of the 19 classes which are all static scenes or objects, while the 4D semantic segmentation task involves the 6 more moving classes leading to a more challenging situation. The temporal information between multiple frames is crucial to obtain better performance on the 4D semantic segmentation task.

Synthia 4D Dataset [11] Synthia [16] is a large synthetic dataset collected for scene understanding, self-driving, and semantic segmentation purpose. It contains more than 200,000 HD images from videos and 20,000 HD images from snapshots under different styles of scenes including European style, modern city, highway, and green areas. The dataset provides groundtruth annotation for 13 class labels of semantic segmentation, depth estimation, and car ego-motion. Recently, MeteorNet [11] creates a Synthia 4D dataset derived from Synthia dataset. They generate 3D videos by back-projecting the depth image to the 3D space. The 6 sequences are selected under 9 weather conditions in different scenarios.

Waymo Open Dataset (WOD) [17] is a recent large-scale self-driving dataset including two datasets including perception and motion dataset. The whole dataset contains 1,150 scenes where the training, validation and testing split consists of 798, 202, and 150 scenes respectively. The perception dataset is annotated with 1,950 lidar sequences while the motion dataset has 103,354 sequences. Each sequence is collected at sampling frequency of 10Hz and last 20s.

3 Point Cloud VS Depth Images

We here compare the data difference between point cloud and depth image. Point clouds are collections of data points in 3D space that can encompass diverse attributes like color and intensity. As shown in Table 2, their flexibility allows them to represent both sparse and dense areas seamlessly, making them versatile for applications ranging from 3D reconstruction to Lidar processing. However, they often require significant memory, especially for dense datasets, and their unordered nature demands specialized algorithms for processing. Additionally, point clouds can sometimes have varying density and accuracy across different sections.

On the other hand, depth images are 2D representations where each pixel's value indicates the distance from the camera to the real-world point. Their structured grid format is beneficial for simplifying many processing tasks, and they maintain a consistent resolution throughout the image. When combined with traditional color cameras, depth images can produce RGB-D data. Despite these advantages, depth images have a fixed memory footprint based on their resolution, mainly provide depth information (unless paired with color), and aren't as adept at representing sparse data compared to point clouds.

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Juergen Gall. SemanticKITTI: A dataset for semantic scene understanding of lidar sequences. In *ICCV*, 2019. 2, 3

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 3

[3] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *CVPR*, 2019. 1

[4] Ayush Dewan, Tim Caselitz, Gian Diego Tipaldi, and Wolfram Burgard. Rigid scene flow for 3d lidar scans. In *IROS*, 2016. 2

[5] Hehe Fan and Yi Yang. PointRNN: Point recurrent neural network for moving point cloud processing. *arXiv preprint arXiv:1910.08287*, 2019. 2

[6] Hehe Fan, Xin Yu, Yuhang Ding, Yi Yang, and Mohan Kankanhalli. Pstnet: Point spatio-temporal convolution on point cloud sequences. In *International Conference on Learning Representations*, 2021. 1

[7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2, 3

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 2, 3

[9] Benjamin Graham. Spatially-sparse convolutional neural networks. *arXiv preprint arXiv:1409.6070*, 2014. 1

[10] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3D: Learning scene flow in 3D point clouds. In *CVPR*, pages 529–537, 2019. 2

[11] Xingyu Liu, Mengyuan Yan, and Jeannette Bohg. MeteorNet: Deep learning on dynamic 3D point cloud sequences. In *ICCV*, 2019. 1, 2, 3

[12] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2016. 2, 3

[13] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015. 2, 3

[14] Gregory P Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K Wellington. Lasernet: An efficient probabilistic 3d object detector for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12677–12686, 2019. 2, 3

[15] Charles R. Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 1

[16] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 3

[17] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 3

[18] Arash K Ushani, Ryan W Wolcott, Jeffrey M Walls, and Ryan M Eustice. A learning approach for real-time temporal scene flow estimation from lidar data. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5666–5673. IEEE, 2017. 2

[19] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 2015. 2