A Dynamic 3D Point Cloud Dataset for Immersive Applications

Yuan-Chun Sun
National Tsing Hua University
Taiwan

I-Chun Huang
National Tsing Hua University
Taiwan

Yuang Shi
National University of Singapore
Singapore

Wei Tsang Ooi
National University of Singapore
Singapore

Chun-Ying Huang
National Yang Ming Chiao Tung University
Taiwan

Cheng-Hsin Hsu
National Tsing Hua University
Taiwan

ABSTRACT

Motion estimation in a 3D point cloud sequence is a fundamental operation with many applications, including compression, error concealment, and temporal upscaling. While there have been multiple research contributions toward estimating the motion vector of points between frames, there is a lack of a dynamic 3D point cloud dataset with motion ground truth to benchmark against. In this paper, we present an open dynamic 3D point cloud dataset to fill this gap. Our dataset consists of synthetically generated objects with pre-determined motion patterns, allowing us to generate the motion vectors for the points. Our dataset contains nine objects in three categories (shape, avatar, and textile) with different animation patterns. We also provide semantic segmentation of each avatar object in the dataset. Our dataset can be used by researchers who need temporal information across frames. As an example, we present an evaluation of two motion estimation methods using our dataset.

CCS CONCEPTS

• **Computing methodologies** → *Volumetric models*; **Virtual reality**; • **Human-centered computing** → **Virtual reality**; • **Information systems** → **Multimedia streaming**.

KEYWORDS

Point cloud, dataset, immersive applications, point matching, register, interpolation, error concealment

ACM Reference Format:
Yuan-Chun Sun, I-Chun Huang, Yuang Shi, Wei Tsang Ooi, Chun-Ying Huang, and Cheng-Hsin Hsu. 2023. A Dynamic 3D Point Cloud Dataset for Immersive Applications. In *Proceedings of the 14th ACM Multimedia Systems Conference (MMSys '23), June 7–10, 2023, Vancouver, BC, Canada.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3587819.3592546

MMSys '23, June 7–10, 2023, Vancouver, BC, Canada

ACM ISBN 979-8-4007-0148-1/23/06…$15.00
https://doi.org/10.1145/3587819.3592546

1 INTRODUCTION

Dynamic 3D point clouds are sequences of point cloud frames along the temporal domain. Each point cloud frame is composed of many unordered points with x, y, and z coordinates and attributes, such as color components and material reflectance. 3D point clouds can be cost-effectively and reliably captured by emerging sensors, including RGBD cameras, 3D scanners, and LiDAR, without: (i) relying on computationally-demanding algorithms or (ii) incurring excessive network traffic. Therefore, different from other 3D representations, like 3D meshes and light field videos, dynamic 3D point clouds have a better chance to be captured, encoded, streamed, and rendered in real-time. The potential of dynamic 3D point clouds has been well-recognized by academia and industry for multiple emerging applications, like Virtual/Augmented Reality (VR/AR), telepresence, heritage digitization, and autonomous driving [23]. The market shares of these applications have grown significantly in the past few years. Such a trend shows no indication of slowing down, e.g., recent market research projects that the VR market will grow from 6.9 billion USD in 2021 to 451 billion in 2030 [37].

To enable and optimize the abovementioned applications, dynamic 3D point clouds need to be analyzed and processed through various algorithms to extract semantic data. These algorithms can be based on traditional signal processing techniques or modern machine learning approaches [43]. Popular algorithms for semantic data include, but are not limited to:

- *Flow estimation* algorithms [8, 23] compute 3D motion vectors for individual points, which can be used for dynamic point cloud compression and streaming.
- *Segmentation* algorithms [49] give one or more class labels to each point of dynamic point clouds. For example, an avatar's head, trunk, and limbs can be segmented for optimization purposes in a telepresence session.
- *Prediction* algorithms [21] generate future point cloud frames based on historical frames, in order to reduce the response time or lower the network traffic amount.

One of the unique challenges when designing the algorithms is the *point matching* problem among the unordered points across multiple point cloud frames. Existing dynamic 3D point cloud datasets [13, 17] do not provide the ground truth of matched points between two distinct frames of the same sequence. This, in turn, makes quantifying the performance of the point matching and

The work was partially supported by the Singapore Ministry of Education Academic Research Fund Tier 1 (T1 251RES2038) and the NSTC of Taiwan (#111-2221-E-007-060).

semantic data extraction algorithms difficult. In fact, multiple research groups resorted to creating synthetic sequences with the ground truth of matched points themselves [22, 42]. Unfortunately, to our best knowledge, their datasets are not public, which renders quickly reproducing and fairly comparing different algorithms tedious, time-consuming, and error-prone, if possible at all.

To cope with the above challenge, we systematically synthesize the *very first* 3D dynamic point cloud dataset with the *ground truth of matched points*, which can be downloaded at http://snoopy.cs.nthu.edu.tw/datasets/PointCloudMMSys23.tar.bz2. To make our own dataset applicable to a wide range of applications, we generate sequences of dense 3D point clouds at high frame rates. We then provide scripts for researchers to downsample our dynamic 3D point cloud sequences in both temporal and spatial domains. To cover heterogeneous 3D scenarios, we consider three object classes: *shape*, *avatar*, and *textile* with diverse characteristics. Each object class comprises three objects. Each object is programmed to move in three movement patterns. In total, the dataset consists of 54 dynamic 3D point cloud sequences, covering both rigid and non-rigid transforms, and 151968 point cloud files.

The resulting dataset can be used to evaluate and compare multiple algorithms, including:

- *Point cloud register* [20, 25, 26] that computes the optimal geometric transformation (e.g., in rotation and translation) to align points in a point cloud frame to another frame. Such problems are critical in domains like robotics and computer vision, where multiple point cloud scans need to be compared and aligned to create a 3D model of a complete environment. Algorithms for registering point clouds are useful to: (i) create virtual worlds in VR/AR applications and (ii) more reliably understand road intersections for autonomous driving.
- *Interpolation* [2, 5, 42, 46] that upsamples dynamic 3D point cloud sequence in the temporal or spatial domain (or both) in order to create more smooth 3D objects or scenes. Algorithms for interpolating 3D point clouds are useful in heritage digitization and autonomous driving.
- *Error concealment* [14, 24, 27] that conceals the distorted point cloud frames after transmission due to lost or late packets in networks. By doing so, the received sequences can be rendered to users at a higher visual quality. Algorithms to conceal transmission errors are useful for distributed VR/AR applications, including telepresence.

The rest of the paper is organized as follows. Sec. 2 surveys relevant media datasets. This is followed by the methodology in Sec. 3. We present the resulting dataset in Sec. 4. Sec. 5 gives sample usages of our dataset. We conclude the paper in Sec. 6.

2 RELATED WORK

We build a dense point cloud dataset for immersive applications. Hence, the current dataset is orthogonal to point cloud datasets collected by LiDAR for autonomous driving [4, 7, 12, 19, 35]. We survey public media datasets related to immersive applications in this section. We classify them into three categories: *2D images*, *3D meshes*, and *3D point clouds* in the following.

2D image datasets can be further classified into *natural* or *synthesized* datasets. For natural datasets, Ofli et al. [33] constructed a multi-modal human dataset consisting of 2D images, multi-view videos, depth videos, human skeletons, acceleration traces, and audio clips of multiple human subjects performing different actions. Similarly, Ganapathi et al. [18] generated a small human dataset of depth videos and human skeletons, while Dai et al. [16] captured a large RGBD image dataset of various scenes with semantic labels. For synthesized datasets, Mayer et al. [31] generated a dataset of flying objects, animated short films, and driving scenarios. Their dataset consists of images, segmentations, and motions. Ros et al. [38] created a dataset of diverse urban images and videos with labels. They simulated different seasons and weather conditions in Unity. Different from the above dataset [16, 18, 31, 33, 38], there are some studies [36, 39, 41] that provided datasets of RGBD images, which can be converted into point clouds. For example, Pumarola et al. [36] constructed a 3D human dataset, which covers a wide array of humans and actions in 3D meshes and textures. Their 3D humans are from Adobe Fuse [1] and MakeHuman [30] and human actions from Mixamo [32]. The resulting 2D videos are rendered in Blender [15]. Varol et al. [41] generated a human action dataset under different lighting conditions, which also contains depth and normals. Shafaei and Little [39] built an RGBD dataset with multiple human poses, which contains point cloud segments. *The aforementioned datasets [16, 18, 31, 33, 36, 38, 39, 41] only contain 2D images, which are different from our dynamic 3D point cloud dataset.*

3D mesh datasets. 3D meshes are widely used in computer graphics, and there exist quite a few public 3D mesh datasets. For example, ShapeNet [11] is a large-scale 3D mesh dataset, which groups 3D models into four categories: single 3D models, 3D scenes, billboards, and big ground planes. These 3D models are classified into 3,135 categories. Xu et al. [45] provided a dynamic human textured mesh sequence dataset. There are four sequences in the dataset, known as "basketball player", "dancer", "exercise", and "model". CLOTH3D [6] is a dataset containing sequences of synthetic 3D human models with cloths. *These datasets [6, 11, 45] contain 3D meshes, which are heavier than 3D point clouds considered by us.*

3D point cloud datasets. We are only aware of five 3D point cloud datasets, which can be categorized into *static* [3, 29, 40, 44] and *dynamic* [13, 17, 28, 34, 47, 48] ones. For static 3D point clouds, Turk and Levoy [40] started to build the Stanford 3D scanning repository since 1994. This repository contains six representative static point cloud objects, such as "Stanford Bunny", which are widely used in recent research. Armeni et al. [3] generated a 3D point cloud dataset of real indoor scenes using a 3D scanner to evaluate their semantic parsing algorithms. Su et al. [29] created a 3D point cloud dataset (WPC Database) of groceries captured by cameras in the real world. The dataset includes 20 objects with diverse geometric and textural complexity levels, and all of the 3D point clouds are of high quality and realistic. More recently, Wu et al. [44] constructed a 3D point cloud dataset using a public 3D mesh dataset to compare the performance of different point cloud compression algorithms. Different from these works, we consider more complex dynamic 3D point clouds. For dynamic point clouds, 8iVFB [17] was a voxelized human dataset of dynamic 3D point clouds. The dataset

consists of four moving humans captured by 42 cameras in 14 clusters. Later in 2018, Krivokuća et al. [28] created a voxelized human dataset (8iVSLF dataset) with much higher resolutions, containing one 300-frame sequence, as well as six high-resolution single-frame point clouds. Zerman et al. created two publicly available volumetric video datasets, i.e., vsenseVVDB [47] and vsenseVVDB2 [48]. VsenseVVDB provides two volumetric videos of football players in colored point cloud format with four different point cloud densities. VsenseVVDB2 provides four new volumetric videos of human avatars in both colored point cloud and textured 3D mesh formats. Pagés et al. [34] introduced the Volograms & V-SENSE Volumetric Video Dataset with three male avatars with varying skin color, clothing, stature, and range of movements. MVUB [13] is another voxelized human dataset with only upper bodies from Microsoft. The dataset is composed of five moving humans, also captured by RGBD cameras. *All the datasets mentioned above focus on humans, and they do not provide the ground truth of matched points.*

3 DATA GENERATION METHODOLOGY

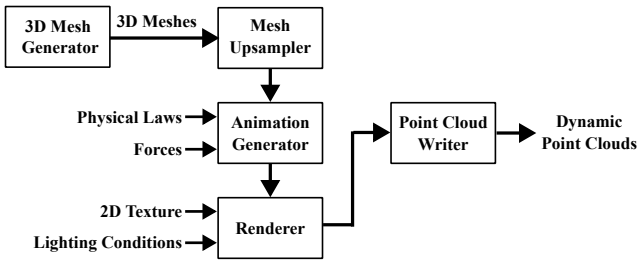


Figure 1: Our data generation pipeline.

In this section, we present our data generation pipeline, which consists of five stages as summarized in Fig. 1.

3.1 Mesh Generator

We generate dynamic 3D point cloud sequences from 3D mesh models. Our dataset consists of three object classes: *shape*, *textile*, and *avatar*. We gather more complex objects from CC0 online resources, like Mixamo [32] and CGTrader [10]. We create simpler ones by ourselves. The resulting 3D mesh models are sent to the next stage.

3.2 Mesh Upsampler

We next upsample 3D mesh models to create dense point clouds. More specifically, we employ two *subdivision* methods on 3D mesh models to increase the number of 3D meshes. The simpler method, known as the *midpoint* method, involves connecting the midpoints of the edges to divide each mesh face into smaller faces. The more complex method is *Catmull-Clark* [9], which generates a smooth surface from a polyhedral mesh of any topology recursively. We employ *midpoint* subdivision for simpler meshes, such as shapes; and *Catmull-Clark* subdivision for complex meshes, such as human skins and wrinkled textiles. The upsampled 3D mesh models have enough meshes for generating dense point clouds.

3.3 Animation Generator

We next create dynamic scenes from static 3D mesh models. In each scene, we select a *main* object from the three object classes (shape, textile, and avatar), along with one or multiple *additional* objects (such as ground, wall, and stairs). The animation generator incorporates physical laws, such as gravity, tension, and elastic coefficients, as well as forces, such as wind and friction forces, on the main objects. For avatars, we adopt human actions, such as walk, kick, and dance, generated by Mixamo [32]. For each scene, we create up to three animations with diverse complexity levels on the movement of its main object. For the avatar class, the objects are segmented based on the semantic of the clothing and the body parts.

3.4 Renderer

The animated scenes of 3D meshes, along with 2D texture and lighting conditions, are rendered into sequences of geometry (coordinates) and attribute (colors) data. We chose the open-source Blender [15] as our 3D mesh model creator, animator, simulator, and renderer. Blender supports two renderers: *Eevee* and *Cycle*. We adopt Cycle for more photorealistic rendering results, at an expense of longer running time. We consider two lighting conditions: *base*, where points are in their original textures and colors without lights, and *lighted*, where points are under lighting conditions. Here, we manually add up to two *sun*- or *spot-lights* for visually appealing rendered results.

3.5 Point Cloud Writer

The output of the renderers is animated 3D meshes with geometry and attribute data. We next extract the geometry and attribute data of individual *vertices* from each animated frame, and turn them into *points* of a sequence point cloud frame. Across the dynamic 3D point cloud frames, we keep track of point indices, which serve as the ground truth of matched points.

4 RESULTING DATASET

We present our dataset in this section.

4.1 3D Objects and Animations

We consider three classes of main objects: *shape* for simple 3D objects for virtual worlds, *avatar* for telecommunications, and *textile* for more challenging setups. In each class, we select three representative mesh objects. For every main object, we generate three animations with diverse complexity levels in terms of movement patterns. To achieve that, several additional objects are added to the animations. Our dataset provides dynamic dense point clouds of both main and additional objects.

The details on all dynamic 3D point cloud sequences can be found in Table 1. Fig. 2 gives sample rendered views of our main objects without and with lights. For *shapes* (Figs. 2(a)–2(c)), we choose a golden *coin*, an unsolved Rubik's *cube*, and a colorful plastic *cup*. Three animations are selected for these three main objects, which are: (i) *freefall* with a *ground* object as an additional object, (ii) *slide* with a 45°*slope* and a ground as additional objects, and (iii) *stair* with *stairs* as additional objects. When creating the animations, we empirically select the initial states (positions and

Figure 2: Rendered views of the considered main objects. All main objects without lights: (a)–(i); sample main objects with lights: (j)–(l).

Table 1: Summary of Dynamic 3D Point Cloud Sequences

Obj.	No. Pt.	Size (cm^3)	Animation	Total Pt.	No. Fme.
Shape					
Coin	114336	8×8×2	Freefall	220472	600
			Slide	269578	600
			Stair	143444	600
Cube	147458	8×8×8	Freefall	212272	600
			Slide	261378	600
			Stair	135244	600
Cup	115586	8×8×10	Freefall	221722	600
			Slide	270828	600
			Stair	144694	600
Avatar					
Man	106810	200×50×200	Walk	106810	600
			Kick	106810	624
			Dance	106810	600
Woman	141269	200×50×200	Walk	141269	600
			Kick	141269	624
			Dance	141269	600
Robot	166150	200×50×200	Walk	166150	600
			Kick	166150	624
			Dance	166150	600
Textile					
Blanket	128018	200×200×0.8	1-Corner	231554	600
			2-Corners	231554	600
			Drop	231554	600
Curtain	131044	100×90×0.5	Breeze	338028	600
			Gale	338028	600
			Drop	338028	600
Flag	128018	150×90×0.2	Breeze	267244	600
			Raise	267244	600
			Wave	267244	600

orientations) of the main objects to create more exciting (complex) movement patterns due to the physical laws. The length of these nine sequences is 600 frames.

For *avatars* (Figs. 2(d)–2(f)), we select a *man* who is a construction worker with a beard wearing a reflective vest and a helmet, a *woman* who wears a white T-shirt and blue jeans, and a *robot* which has smooth shells and round joints. Three animations are chosen for these three main objects, while no additional objects are added. The three animations, from simple to complex, are: (i) *walk*, in which an avatar walks in a circle with a swinging right arm and fixed left arm (as holding an object), (ii) *kick*, in which an avatar performs six different kicks, and (iii) *dance*, in which an avatar dances in a hip-hop style featuring complex transitions. The length of these nine sequences is at least 600 frames.

For the most complicated *textiles* (Figs. 2(g)–2(i)), we pick a red squared cotton *blanket* with white grid patterns, window *curtains*, and a blue *flag*. We create diverse animations for different textile objects, as detailed below.

- *Drag a corner of the blanket (1-Corner).* We grab a corner of the blanket, place it on top of a static robot (one of the avatars above), and drag the blanket in various directions.
- *Drag two corners of the blanket (2-Corners).* We grab and drag two adjacent corners of the blanket, also on a static robot.
- *Falling blanket (Drop).* The blanket falls vertically on a crouched robot, which then stands up.
- *Blowing curtains (Breeze)* We add the ground, a wall, and a window as additional objects for curtains. We open the curtains under a gentle breeze, where random gusts are added to make the curtains move irregularly.
- *Blowing curtains (Gale).* Similar to a curtain blowing in a breeze, but with a wind speed that is five times faster.
- *Falling curtains (Drop).* Similar to the gale, but detach the curtains in the middle of the sequence.
- *Blowing flag (Breeze).* We add the ground and a flag pole as additional objects. We attach the flag at the top of the pole, under a breeze with some random gusts.
- *Raise the flag (Raise).* We raise and then lower the flag.
- *Blowing flag (Wave).* We emulate the movement pattern of an invisible hand waving a flag.

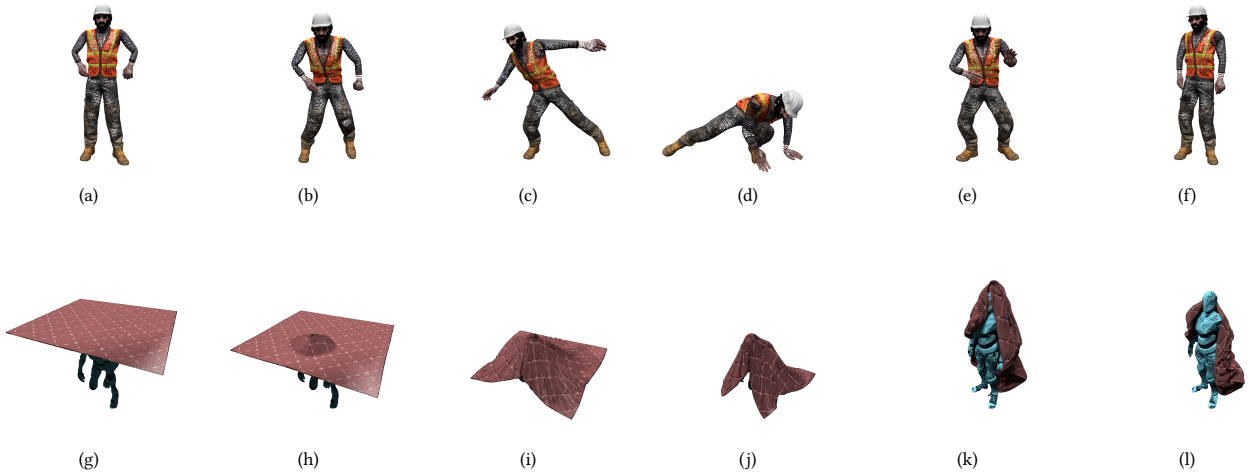


Figure 3: Sample rendered animations: dancing man: (a)–(f), and dropping blanket: (g)–(l).

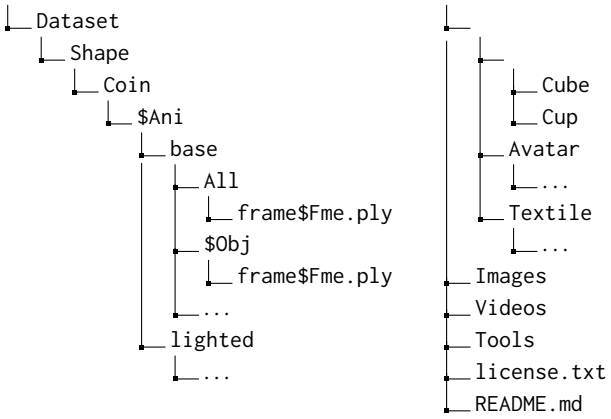


Figure 4: Directory structure.

Each of these nine animations lasts for 600 frames. We present two sample animations in Fig. 3: a dancing man and a dropping blanket.

4.2 Dataset Files

The directory structure is shown in Fig. 4. For each animation $Ani of a main object, we provide two versions of the dynamic 3D point cloud sequences: base and lighted. Each animation may contain multiple objects, and we provide sequences of individual objects $Obj, such as coin, slope, and ground. In addition, we merge these dynamic 3D point clouds into an *all* dynamic 3D point cloud sequences for the convenience of dataset users. We save the 3D point cloud frames in ASCII `ply` files, frame by frame, where the filenames have a suffix of their 1-based frame numbers $Fme. Note that, by providing a separate ply file for each object, our dataset offers simple labels for semantic segmentation. Such labels could be particularly useful for avatar objects. Take woman as an example, its 3D points are labeled with: body, eyelashes, hair, shirt, pants, and shoes.

We also release the following three tools for researchers and practitioners to use our dataset:

- *Temporal downsampling tool* that downsamples the frames to a specified number of frames by repeatedly skipping a few consecutive point cloud frames.
- *Spatial downsampling tool* that downsamples the number of points from individual 3D point cloud frames. We use an efficient downsampling method, i.e., *random downsampling*, to downsample the point cloud frame without destroying the ground truth of matched points.
- *Object scaling tool* that adjusts the size of point cloud objects. This tool scales the object with the origin (0,0,0) as the center.

The scripts can be found under the "/Tools" folder in our dataset.

4.3 Generating and Rendering Animations

We generated the dataset on a workstation with an Intel Core i9-9920X CPU at 3.5GHz and NVIDIA GeForce RTX 3080 Ti GPU. It took us 30+ hours to generate animations of each main object, which included the following steps: searching for 3D models, pre-processing the 3D models, adding the animations, setting the light conditions, and rendering the animations. Take the Curtain object as an example; it took us at least one hour to simulate the curtain movement of 600 frames, three hours on average to render each animation, and six hours on average to convert all intermediate data files into dynamic 3D point cloud sequences.

5 USEFULNESS OF OUR DATASET: POINT MATCHING AS A SHOWCASE

Our dataset can be used in multiple applications that temporarily crunch 3D point cloud frames of a sequence. The crux of optimizing these applications (see Sec. 1) is the quality of point matching results. In this section, we employ point matching to showcase

how researchers and practitioners can use our dataset. In general, without our dataset, they can first analyze the characteristics of dynamic point clouds and then design/optimize their algorithms or models based on the observations.

Point matching problem. Given a previous frame f_p and a future frame f_n with the same number of points, we want to find the one-to-one matching between points $p \in f_p$ and $q \in f_n$. Here, q represents the ground-truth mapping in our dataset. Let q' denote the matched point of p generated by an algorithm under evaluations, we can formally define the matching errors of q' as:

$$d_g(q, q') = \sqrt{(q.x - q'.x)^2 + (q.y - q'.y)^2 + (q.z - q'.z)^2}; \quad (1)$$

$$d_r(p, q, q') = \cos^{-1}\left(\frac{(q-p)\cdot(q'-p)}{d_g(q,p) \times d_g(q',p)}\right); \quad (2)$$

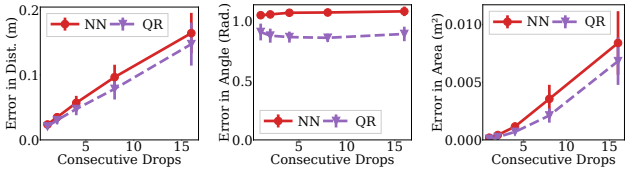
$$d_a(p, q, q') = (q-p) \times (q'-p), \quad (3)$$

where $d_g(\cdot), d_r(\cdot), d_a(\cdot)$ are deviations between q' and q in Euclidean distance, angle, and area, respectively. Although other error metrics, e.g., those that also take RGB colors into account, can also be defined, we consider these three error metrics for brevity in this section. Their units are m, radian, and m^2. The goal of the problem at hand is to minimize the overall error across all $q' \in f_n$.

(a) (b) (c)

Figure 5: Matched points of kicking woman, where lines connect the matched points in f_p and f_n: (a) ground truth, (b) NN, and (c) QR. Only 2% of lines are shown for clarity.

Qualitative comparison. In our earlier work [27], we developed several algorithms to conceal a missing point cloud frame based on a previous and a future frame: f_p and f_n. Two point matching approaches were proposed: (i) Nearest Neighbor (NN), in which the point $q' \in f_n$ that has the smallest Euclidean distance to $p \in f_p$ is selected, and (ii) Query Radius (QR), in which a point $q' \in f_n$ with the highest similarity function value is chosen from all candidate points within a radius τ to $p \in f_p$. Upon points in the previous and future frames being matched, the missing point cloud frame is concealed with a suite of tools in our proposed pipeline. To qualitatively compare the matching quality of NN and QR, we randomly drop a frame from the kicking woman sequence and execute NN and QR to match points in the preceding and following frames. We connect the matched points in Fig. 5 based on the ground truth and results from NN and QR. Compared to Fig. 5(a), NN and QR result in multiple *mismatched* points in both *direction* and *magnitude*. Fig. 5(b) shows that NN matches the woman's left calf and foot with her left thigh; and her fingertips with palms. Fig. 5(c) reveals that QR occasionally matches points from her left foot to her right thigh, which may be caused by a rather large τ value.



(a) (b) (c)

Figure 6: Error comparison between NN and QR under different numbers of consecutive drops: (a) distance, (b) angle, and (c) area. Sample results from the dancing man are shown.

Quantitative comparison. Next, we quantify the performance of NN and QR using the error metrics in Eqs. (1)–(3). We select 50 consecutive frames from the dancing man sequence for the experiments. We vary the number of consecutive frame drops from 1 to 16 and repeatedly drop the frames. We then run NN and QR[1] to match points between f_p and f_n. We report the average errors with 95% confidence intervals in Fig. 6. Figs. 6(a) and 6(c) show that NN and QR perform equally well in distance and area if the number of consecutive drops is small (e.g., under good network conditions), while their gaps dramatically increase as the number of consecutive drops increases (e.g., under network congestion). In contrast, Fig. 6(b) reveals that the difference between NN and QR in angle is rather stable across different numbers of consecutive drops.

We emphasize that Figs. 5 and 6 along with qualitative and quantitative comparisons would not be possible without the point matching ground truth provided by our dataset. Multiple recommendations for Hung et al. [27] can be drawn, e.g., the search radius τ should be adaptive in both temporal and spatial domains; and point matching algorithms can be optimized given specific movement patterns. Last, we emphasize that point matching algorithms are critical to error concealment of dynamic 3D point clouds as well as other machine learning algorithms. Developers of those algorithms will find our dataset valuable.

6 CONCLUSION

In this paper, we presented the first 3D dynamic point cloud dataset with the point-matching ground truth. In this dataset, we selected three object classes that contain a total of nine objects and 27 animations to generate the high-resolution and high frame-rate point cloud sequences. We also made some extra efforts to provide point cloud sequences under some lighting conditions and perform simple semantic segmentation, labeling the avatars' clothing and body parts. We show how our dataset can benefit the previous research, which lacks ground truth point-matching. The generated dataset can be utilized by a wide range of individuals, including researchers, engineers, and hobbyists, throughout various stages of development, such as design, fine-tuning, and evaluations. Not only the temporal information (motion estimation) can be studied, the semantic segmentation can also be explored in our future work.

[1]A grid search on the best τ value is carried out. Detail is omitted for brevity.

REFERENCES

[1] Adobe. 2023. *Adobe Fuse.* https://rotf.lol/5n7ekucm(https://rotf.lol/5n7ekucm).

[2] Anique Akhtar, Zhu Li, Geert Van der Auwera, and Jianle Chen. 2022. Dynamic Point Cloud Interpolation. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Singapore, Singapore, 2574–2578.

[3] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 2016. 3D Semantic Parsing of Large-Scale Indoor Spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, 1534–1543.

[4] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. 2019. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea, 9297–9307.

[5] Matthew Berger, Andrea Tagliasacchi, Lee M Seversky, Pierre Alliez, Gael Guennebaud, Joshua A Levine, Andrei Sharf, and Claudio T Silva. 2017. A survey of surface reconstruction from point clouds. In *Computer Graphics Forum*. Los Angeles, CA, 301–329.

[6] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. 2020. CLOTH3D: Clothed 3D Humans. In *Proc. of Springer International Publishing Computer Vision – ECCV 2020*. Cham, Germany, 344–359.

[7] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, 11621–11631.

[8] Chao Cao, Marius Preda, and Titus Zaharia. 2019. 3D Point Cloud Compression: A Survey. In *The 24th International Conference on 3D Web Technology*. New York, NY, 1–9.

[9] Edwin Catmull and James Clark. 1978. Recursively generated B-spline surfaces on arbitrary topological meshes. *Elsevier Computer-aided design* 10, 6 (1978), 350–355.

[10] CGTrager. 2023. *CGTrader.* https://www.cgtrader.com/(https://www.cgtrader.com/).

[11] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. 2015. ShapeNet: An Information-Rich 3D Model Repository. *CoRR* abs/1512.03012 (2015).

[12] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. 2019. Argoverse: 3D Tracking and Forecasting With Rich Maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA.

[13] L Charles, Q Cai, E Sergio Orts, and A Chou Philip. 2021. JPEG pleno database: microsoft voxelized upper bodies-a voxelized point cloud dataset.

[14] Jingdao Chen, John Seon Keun Yi, Mark Kahoush, Erin S Cho, and Yong K Cho. 2020. Point Cloud Scene Completion of Obstructed Building Facades with Generative Adversarial Inpainting. *Multidisciplinary Digital Publishing Institute Sensors* 20, 18 (2020), 5029.

[15] Blender Online Community. 2018. *Blender - a 3D modelling and rendering package.* http://www.blender.org(http://www.blender.org).

[16] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. 2017. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, 5828–5839.

[17] Eugene d'Eon, Bob Harrison, Taos Myers, and Philip A Chou. 2017. 8i voxelized full bodies-a voxelized point cloud dataset. *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006* 7, 8 (2017), 11.

[18] Varun Ganapathi, Christian Plagemann, Daphne Koller, and Sebastian Thrun. 2012. Real-Time Human Pose Tracking from Range Data. In *Proc. of Springer Berlin Heidelberg Computer Vision – ECCV 2012*, Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid (Eds.). Berlin, Heidelberg, 738–751.

[19] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. Providence, RI, 3354–3361.

[20] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. 2019. The perfect match: 3d point cloud matching with smoothed densities. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. Long Beach, CA, 5545–5554.

[21] Pedro Gomes, Silvia Rossi, and Laura Toni. 2021. Spatio-Temporal Graph-RNN for Point Cloud Prediction. In *Proc. of IEEE International Conference on Image Processing (ICIP'21)*. Anchorage, AK, 3428–3432.

[22] Pedro Gomes, Silvia Rossi, and Laura Toni. 2021. Spatio-temporal Graph-RNN for Point Cloud Prediction. In *2021 IEEE International Conference on Image Processing (ICIP)*. Anchorage, AK, 3428–3432.

[23] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai. 2020. An overview of ongoing point cloud compression standardization activities: video-based (V-PCC) and geometry-based (G-PCC). *APSIPA Transactions on Signal and Information Processing* 9 (2020), e13.

[24] Ju He, Zeqing Fu, Wei Hu, and Zongming Guo. 2019. Point cloud attribute inpainting in graph spectral domain. In *2019 IEEE International Conference on Image Processing (ICIP)*. Taipei, Taiwan, 4385–4389.

[25] Jing Huang and Suya You. 2012. Point cloud matching based on 3D self-similarity. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. Providence, RI, 41–48.

[26] Xiaoshui Huang, Guofeng Mei, Jian Zhang, and Rana Abbas. 2021. A comprehensive survey on point cloud registration. *arXiv preprint arXiv:2103.02690* (2021).

[27] Tzu-Kuan Hung, I-Chun Huang, Samuel Rhys Cox, Wei Tsang Ooi, and Cheng-Hsin Hsu. 2022. Error Concealment of Dynamic 3D Point Cloud Streaming. In *Proceedings of the 30th ACM International Conference on Multimedia*. Lisbon, Portugal, 3134–3142.

[28] Maja Krivokuća, Philip A. Chou, and Patrick Savill. 2018. 8i Voxelized Surface Light Field (8iVSLF) Dataset. *ISO/IEC JTC1/SC29 WG11 (MPEG) input document m42914* (2018).

[29] Qi Liu, Honglei Su, Zhengfang Duanmu, Wentao Liu, and Zhou Wang. 2022. Perceptual Quality Assessment of Colored 3D Point Clouds. *IEEE Transactions on Visualization and Computer Graphics* (2022), 1–1.

[30] Makehuman. 2023. *MakeHuman.* https://rotf.lol/3k47ed8x(https://rotf.lol/3k47ed8x).

[31] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. 2016. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV.

[32] Mixamo. 2023. *Adobe's Mixamo.* https://www.mixamo.com(https://www.mixamo.com).

[33] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy. 2013. Berkeley MHAD: A comprehensive Multimodal Human Action Database. In *2013 IEEE Workshop on Applications of Computer Vision (WACV)*. Clearwater Beach, FL, USA, 53–60.

[34] Rafael Pagés, Konstantinos Amplianitis, Jan Ondrej, Emin Zerman, and Aljosa Smolic. 2021. Volograms & V-SENSE Volumetric Video Dataset. *ISO/IEC JTC1/SC29/WG07 MPEG2021/m56767* (2021).

[35] Yancheng Pan, Biao Gao, Jilin Mei, Sibo Geng, Chengkun Li, and Huijing Zhao. 2020. SemanticPOSS: A Point Cloud Dataset with Large Quantity of Dynamic Instances. In *2020 IEEE Intelligent Vehicles Symposium (IV)*. Las Vegas, NV, 687–693.

[36] Albert Pumarola, Jordi Sanchez-Riera, Gary P. T. Choi, Alberto Sanfeliu, and Francesc Moreno-Noguer. 2019. 3DPeople: Modeling the Geometry of Dressed Humans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea.

[37] ReportLinker. 2022. *Virtual Reality Market Size, Share and Trends Analysis Report by End-User Type, Product Type and Region, 2021-2030.* https://tinyurl.com/ywvam577 (https://tinyurl.com/ywvam577).

[38] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. 2016. 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 (CVPR)*. Las Vegas, NV.

[39] Alireza Shafaei and James J. Little. 2016. Real-Time Human Motion Capture with Multiple Depth Cameras. In *2016 13th Conference on Computer and Robot Vision (CRV)*. Victoria, BC, Canada, 24–31.

[40] Greg Turk and Marc Levoy. 1994. Zippered polygon meshes from range images. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. 311–318.

[41] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. 2017. Learning From Synthetic Humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI.

[42] Irene Viola, Jelmer Mulder, Francesca De Simone, and Pablo Cesar. 2019. Temporal Interpolation of Dynamic Digital Humans using Convolutional Neural Networks. In *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. San Diego Bay, CA, 90–907.

[43] Haiyan Wang and Yingli Tian. 2022. Sequential Point Clouds: A Survey.

[44] Cheng-Hao Wu, Chih-Fan Hsu, Tzu-Kuan Hung, Carsten Griwodz, Wei Tsang Ooi, and Cheng-Hsin Hsu. 2022. Quantitative Comparison of Point Cloud Compression Algorithms with PCC Arena. *IEEE Transactions on Multimedia* (February 2022), 1–16.

[45] Yi Xu, Yao Lu, and Ziyu Wen. 2017. Owlii Dynamic human mesh sequence dataset. *ISO/IEC JTC1/SC29/WG11 m41658* (2017).

[46] Yiming Zeng, Yue Qian, Qijian Zhang, Junhui Hou, Yixuan Yuan, and Ying He. 2022. 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)*. New Orleans, LA, 6338–6347.

[47] Emin Zerman, Pan Gao, Cagri Ozcinar, and Aljosa Smolic. 2019. Subjective and objective quality assessment for volumetric video compression. *Electronic Imaging* 2019, 10 (2019), 323–1.

[48] Emin Zerman, Cagri Ozcinar, Pan Gao, and Aljosa Smolic. 2020. Textured mesh vs coloured point cloud: A subjective study for volumetric video compression.

In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 1–6.

[49] Jiaying Zhang, Xiaoli Zhao, Zheng Chen, and Zhejun Lu. 2019. A Review of Deep Learning-Based Semantic Segmentation for Point Cloud. *IEEE Access* 7 (December 2019), 179118–179133.