A Comparative Study of K-Planes vs. V-PCC for 6-DoF Volumetric Video Representation

Na Li
Rutgers University
Piscataway, NJ, USA
na.li@rutgers.edu

Mufeng Zhu
Rutgers University
Piscataway, NJ, USA
mz526@rutgers.edu

Shuoqian Wang
SUNY Binghamton
Binghamton, NY, USA
swang130@binghamton.edu

Yao Liu
Rutgers University
Piscataway, NJ, USA
yao.liu@rutgers.edu

ABSTRACT

With NeRF, neural scene representations have gained increased popularity in recent years. To date, many models have been designed to represent dynamic scenes that can be explored in 6 degrees-of-freedom (6-DoF) in immersive applications such as virtual reality (VR), augmented reality (AR), and mixed reality (MR). In this paper, we aim to evaluate how newer neural representations of 6-DoF video compare with more-traditional point cloud-based representations in terms of their representation and transmission efficiency. We design a new methodology for fair comparison between `K-Planes`, a new dynamic neural scene representation model, and video-based point cloud compression (`V-PCC`). We conduct extensive experiments using three datasets with a total of 11 sequences with different characteristics. Results show that the current `K-Planes` models excel for moderately dynamic content, but struggle with highly dynamic scenes. In addition, in emulated volumetric data capture scenarios, the recorded point cloud data can be highly noisy, and the visual quality of views rendered by trained `K-Planes` models are significantly better than `V-PCC`.

CCS CONCEPTS

• **Information systems → Multimedia streaming**; • **Computing methodologies → Point-based models**; **Volumetric models**.

KEYWORDS

6-DoF, volumetric videos, point cloud, neural scene representations

ACM Reference Format:
Na Li, Mufeng Zhu, Shuoqian Wang, and Yao Liu. 2024. A Comparative Study of K-Planes vs. V-PCC for 6-DoF Volumetric Video Representation. In *16th International Workshop on Immersive Mixed and Virtual Environment Systems (MMVE '24), April 15–18, 2024, Bari, Italy*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3652212.3652227

1 INTRODUCTION

In recent years, both the ubiquity and sophistication of devices for video collection have grown. Concurrently, the capabilities of neural network models for fusing information from multiple video signals have seen substantial growth. These two general developments set the stage for more-immersive multimedia streaming applications aimed at enhancing user experiences. Among immersive applications, 6-DoF volumetric video, which captures a real-world scene from a multitude of perspectives over time, enables the greatest level of immersion. Traditional 6-DoF representations include 3D triangular meshes with texture and point clouds. These representations rely on more-direct storage and rendering of 3D scenes. On the other hand, neural representations are often implicit representations of a scene: the stored data powering the representation is often not interpretable by humans. For example, NeRF [23] uses a single multi-layer perceptron (MLP) for representing a scene. Both generating such representations by training from collected imagery and rendering these representations can require significant computational resources.

Although many lines of research have explored 6-DoF representations for individual scenes, addressing the additional temporal dimension in 6-DoF videos presents more challenges. Traditional 6-DoF representations can be adapted for video transmissions by transmitting standard video-encoded streams of RGB-Depth data, as demonstrated in prior work such as [17]. Alternatively, representation-specific codecs [22, 27, 29] such as video-based point cloud compression (`V-PCC`) [26, 28] and Draco [9, 18] can be employed. However, neural representations for 6-DoF video are less-well explored. These representations must simultaneously capture temporal and spatial characteristics and also allow for space-efficient network transmission and compute-efficient rendering.

In this paper, we set to evaluate how newer neural representations of 6-DoF videos compare with more-traditional point cloud-based representations in terms of their representation and transmission efficiency. For our comparison, we use the `K-Planes` model [14] as the state-of-the-art approach for future neural 6-DoF video representation that can be efficiently transmitted. This model strikes a balance between space and computational efficiency, using a representation that factors over both space and time dimensions coupled with a small neural network. For traditional point-cloud-based representations, we select `V-PCC` as it is an emerging standard for compressing dynamic point cloud data.

To perform fair comparison, we design a new methodology including the generation of training data for K-Planes and testing data for both K-Planes and V-PCC as well as the implementation of experiment procedures. Our study uses three different datasets of dynamic 6-DoF scenes. Among them, two are derived from existing datasets, while the third has been created by our team. We have conducted extensive experiments across these three datasets with 11 dynamic sequences with different characteristics. To the best of our knowledge, we are both the first to propose such a comparison methodology and the first to present results from such a comparison study of 6-DoF video representations. The configuration files used for K-Planes training in our experiments along with the trained models are available at: https://github.com/symmru/MMVE-2024.

Results show that for dynamic 6-DoF content with little to moderate motion, using K-Planes models for representation can save the storage size and improve visual quality of rendered views compared to using V-PCC -based encoding. However, the current K-Planes models cannot represent highly dynamic content very well. Moreover, in a emulated real-world scenario where point cloud data is derived from recorded RGB and depth information, we find that the derived point cloud data is very noisy. This confirms the insights from previous studies, e.g., [19, 20]. The visual quality of V-PCC suffers significantly. On the other hand, neural-based solution K-Planes performs substantially better compared to V-PCC in such emulated scene capture scenario.

2 BACKGROUND AND RELATED WORK

Traditional 6-DoF representations. Volumetric videos capture frame sequences in a 3D space, allowing users to view in 6 degree-of-freedom (6-DoF): from arbitrary positions, (x, y, z), in 3D space and arbitrary orientations, (ϕ, θ, ρ). 6-DoF content is widely employed in today's computer gaming and virtual reality platforms. In these platforms, objects and scenes are represented as synthetic models using 3D triangular meshes with texture information that describes how faces of the mesh should appear. Besides trianglular meshes, another volumetric video representation, **point clouds**, has received increased interests in recent years. Point clouds associate color information with 3D pixel/point positions. They can be captured from real-world scenes using RGB-Depth cameras. Typical point cloud scenes contain millions of points and are infeasible to store in raw formats. Point cloud compression (PCC) is currently under active development under Moving Picture Experts Group (MPEG). Among the efforts, video-based point cloud compression (**V-PCC**) [15, 26] aims to leverage existing 2D video codecs for compressing dense point cloud data.

NeRF-based neural representations. Neural radiance field (NeRF) is an emerging representation of 3D scenes. It uses the volume rendering technique for rendering color of pixels on an image. To render a view of a scene, rays are traced from the camera origin through each pixel in the rendered image. The original NeRF [23] proposes to use a simple multi-layer perceptron (MLP) to estimate the volume density σ_i and color $\mathbf{c}_i$ of sample i on a ray as a function of its position $\mathbf{x}_i$ and direction of the ray $\mathbf{d}_i$. The training time of the original NeRF is known to be very long. The authors described in their paper that a typical training can take 1 to 2 days on a Nvidia V100 GPU.

TensoRF [11] is a more recent work that represents the radiance field as a 4D tensor. The main idea of TensoRF is to use tensor decomposition to represent the 4D tensor as the sum of vector-matrix outer products. Compared to the original NeRF, TensoRF models can be trained substantially faster (more than 100x improvement) and with better rendering quality.

Neural representations for dynamic 6-DoF content. For modeling dynamic scenes, many NeRF-variants exist, e.g., D-NeRF [24] and DyNeRF [21]. Among them, K-Planes [14] is a novel approach that represents dynamic volumetric content as a 4D volume (as opposed to the static 3D volume). K-Planes factorizes a 4D volume into 6 planes: 3 space-only planes and 3 space-time planes. Given $q = (i, j, k, t)$ on the 4D volume, it is projected onto each of the 6 planes and bilinearly interpolated to obtain 6 feature vectors. Features from all 6 planes are combined using the Hadamard product (elementwise multiplication). Additionally, K-Planes uses multi-scale planes with different resolutions. Features obtained from different scales $s \in S$ are concatenated. To determine the density and color, it uses two MLPs. The first MLP $\mathcal{F}_\sigma$ is for mapping the feature into volume density σ and an additional feature $\hat{f}(q)$. The second MLP $\mathcal{F}_c$ estimates the color using the additional feature $\hat{f}(q)$ and input of ray direction $\mathbf{d}$.

3 METHODOLOGY

To compare the performance of V-PCC and K-Planes for 6-DoF video representation, we use three datasets with a total of 11 dynamic volumetric sequences. We next describe details of our methodology for conducting this comparative analysis, including the generation of datasets generation as well as the selection of metrics used for comparison.

3.1 Datasets

We use three datasets in this study. The front-facing images of all 11 sequences in the three datasets are shown in Figure 1. The 8iVFB dataset [13] consists of four dynamic point cloud sequences, Longdress , Loot, Soldier, and Redandblack as shown in Figure 1(a)-(d). It is a voxelized full body dataset, with the spatial resolution of each sequence being 1024x1024x1024. The vsenseVVDB2 dataset [31] also includes four dynamic point cloud sequences, AxeGuy, LubnaFriends, Rafa2, and Matis. Similar to 8iVFB , the spatial resolution of sequences in vsenseVVDB2 is also 1024x1024x1024.

Both 8iVFB and vsenseVVDB2 are datasets created for evaluating the performance of V-PCC . They only contain raw points data for each sequence. To use these datasets for comparable K-Planes evaluation, we must generate camera views from different perspectives with known camera extrinsic parameters. Unfortunately, neither of these datasets provide the raw camera-captured video frame data. To obtain data for training the K-Planes models, we use Blender 3.5.1 [5] to render the raw point clouds and use them as the groundtruth data. We describe how we generate training data for K-Planes in Section 3.1.1.

The third dataset Blender is created by our team. It includes three animated Blender 3D models, Lego [23], Pig downloaded from Blender Market [7], Amily downloaded from Blender Demo Files [6]. For the "Lego" model, we created animation raising and lowering the bulldozer's bucket by moving the control panel built

(a) Longdress (b) Loot (c) Soldier (d) Red&Black (e) AxeGuy (f) Lubna (g) Rafa2 (h) Matis (i) Lego (j) Pig (k) Amily

Figure 1: Front-facing images all 11 testing traces. (a)-(d): `8iVFB` dataset; (e)-(h): `vsenseVVDB2` dataset; (i)-(k): `Blender` dataset. All traces contain dynamic sequences that can be explored in 6-DoF.

in original `.blend` file. For "Pig" and "Amily" models, we used the animations in the downloaded `.blend` file. Since `V-PCC` only takes point cloud data `.ply` as input, for fair comparisons, we also need to convert these three models in the `Blender` dataset into point clouds. We emulate the process of capturing real-world point clouds via RGB-Depth cameras within the Blender environment. We must also note, however, that despite accurate camera intrinsics and extrinsics data provided by Blender, due to other factors such as depth data quantization, the recorded depth data is inherently noisy, as with real-world LiDAR sensors [19, 20].

3.1.1 `K-Planes` *Training Data Generation.* For all three datasets, we place the center of the model (`.blend` model or the imported point cloud) at the origin (0,0,0). We follow the original NeRF work [23] and generate `K-Planes` training data by placing virtual cameras in the scene at 80 different positions, starting at position (0,4.0,0.5), which is approximately 4.03 units away from the origin. All 80 positions are obtained by rotation around the origin by a set of randomly generated Euler angles. At any position, the virtual camera is set to "look at" the origin. Since 9 out of 11 sequences are persons, we limit the camera positions to the upper hemisphere only. Note that with this setup, regardless of the camera positions, the distance from the origin is not changed. For each frame in a dynamic sequence, we render 80 views as recorded by 80 virtual cameras with a resolution of 800x800 and save them as `.png` files.

3.1.2 Testing Data Generation. For comparing the visual quality of rendered views, we generate ground truth testing data in a similar way as `K-Planes` training data generation. For the `8iVFB` and `Blender` datasets, we use 20 views from 20 differently-positioned cameras for testing. For the `vsenseVVDB2` dataset, since we use all 300 frames of the dataset, we use views from 10 different perspectives for evaluation. Besides, we set the same random seed for each dataset to make sure all consecutive frames of each model have the same camera parameters.

3.2 Comparison Metrics

To characterize the performance of different codecs, the video compression community commonly uses the rate-distortion (RD) curve e.g., [16, 30]. Here, "rate" represents the bitrate of the encoded media content. "Distortion" represents the visual quality of the compressed representation compared to the ground truth, uncompressed, representation. In this work, we focus on two distortion metrics: peak signal-to-noise ratio (PSNR) and video multi-method assessment fusion (VMAF) proposed by Netflix [1].

To plot the RD-curve for `V-PCC` , we use five different `qp` combinations described in common test conditions (CTC) by MPEG [25]. Details of the five settings are shown in Table 1. Among the five

Table 1: `qp` combinations used in `V-PCC` common test conditions (CTC) [25]

qp settings	r1	r2	r3	r4	r5
Q_g: qp for geometry map	32	28	24	20	16
Q_c: qp for attribute (color) map	42	37	32	27	22

Table 2: `K-Planes` overall settings

Multi-scale	S=1,2; S=1,2,4; S=1,2,4,8
Time dimension	30; 60; 75
Feature length	F=4; F=8; F=16; F=32

configurations, `r1` and `r5` result in the lowest and highest bitrates, respectively.

`K-Planes` uses multi-scale planes with different resolutions for storing parameters. Following the setup in the `K-Planes` paper [14], we consider four spatial scale settings, {1, 2, 4, 8}. With different spatial scale settings, the resolution of the feature plane differs. For example, with $S = 1$, each spatial feature plane has the resolution of 64×64, and the scene contains 64^3 voxels. With $S = 8$, the resolution of the spatial feature plane is 512×512. To inference the density and color of a sample on a ray, features obtained from multi-scale planes are concatenated before being passed into the MLPs. In our experiments, we consider 3 different multi-scale settings, as shown in Table 2. In addition, we consider the impact of setting the time dimension to different values for representing the 4D volume. For feature vector at a plane position, we consider four different feature length settings: 4, 8, 16, and 32.

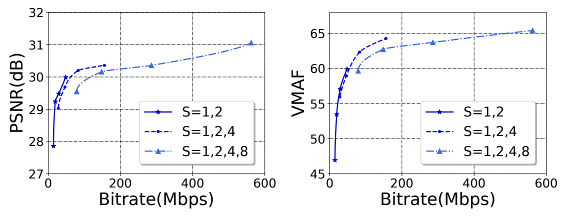
The RD-curve allows us to calculate the average difference in bitrates among different encoding mechanisms under the same distortion. This metric is called the Bjøntegaard-Delta bitrate (BD-rate) [3, 4, 10]. A negative BD-rate represent bitrate/bandwidth savings while achieving the same visual quality and is thus considered better. Similarly, a BD-PSNR metric can be calculated, where a positive number represents the improvement in PSNR while using the same bitrate/bandwidth. We report numerical results of the following metrics: BD-PSNR, BD-Rate$_p$ calculated using PSNR as the visual quality metric; BD-VMAF, and BD-Rate$_v$ calculated using VMAF.

4 K-PLANES RESULTS

In this section, we first characterize the performance of K-Planes for dynamic 6-DoF video representation under different model configurations. Specifically, we compare three multi-scale settings as listed in Table 2 and two time dimension settings.

4.1 Multi-Scale Settings

Figure 2 shows the the RD-curve results, using PSNR and VMAF as the visual quality metric, for the "Lego" sequence in the `Blender`



(a) PSNR (b) VMAF

Figure 2: RD-curve result for the "Lego" sequence when using different numbers of spatial plane scales.

Table 3: BD-Rate⇓, BD-PSNR⇑, BD-VMAF⇑ results on scale performance, using S=1,2 as the anchor for calculation.

Settings	BD-Rate$_p$⇓	BD-PSNR⇑	BD-Rate$_v$⇓	BD-VMAF⇑
S=1,2	0%	0	0%	0
S=1,2,4	37.6%	-0.29	15.7%	-0.94
S=1,2,4,8	122.4%	-1.09	68.1%	-2.32

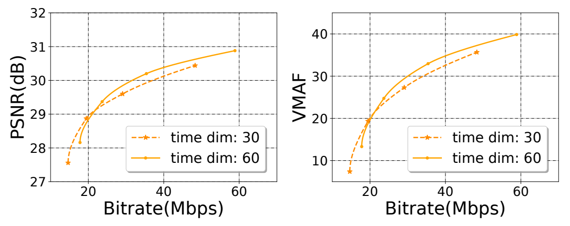
dataset. This sequence has 60 frames. We set the time dimension of the model to 30 (half of the number of frames as used in `K-Planes` [14]). The figure shows three curves, representing the RD-curve for multi-scale settings S=1,2; S=1,2,4; S=1,2,4,8, respectively. For each curve, we vary the "rate" by using different feature length $F \in \{4, 8, 16, 32\}$ for the model. Overall, 12 `K-Planes` models are trained, each using 40 training videos with a learning rate of 0.001. Following the configuration file used for `K-Planes` dynamic scene training [14], we set the number of the training epochs to 120, 000. For each `K-Planes` model, 20 testing videos are used for visual quality evaluation. We calculate the "rate" by considering the frame rate of the sequence to be 30 frames-per-second (fps). That is, 2 seconds to playback 60 frames in the sequence.

The RD-curves confirm that with larger feature length (thus more parameters), the visual quality consistently improves. We further analyzed the BD-Rate, BD-PSNR, and BD-VMAF results in Table 3. Note here that BD-Rate$_p$ is calculated using PSNR as visual quality, while BD-Rate$_v$ is calculated using VMAF. We use S=1,2, the smallest multi-scale setting as the anchor setting for calculating BD-* results. Results show that neither S=1,2,4 nor S=1,2,4,8 can outperform the smallest multi-scale setting, S=1,2. Their BD-Rates with respect to S=1,2 are positive, indicating more bitrates are needed to reach the same visual quality; and their BD-PSNR and BD-VMAF results are negative, indicating worse visual quality under the same bitrates. Based on these findings, we focus the remaining experiments of `K-Planes` on the S=1,2 multi-scale setting.

4.2 Time Dimension Settings

For representing a dynamic 3D scene, `K-Planes` includes six planes, three space-only planes and three space-time planes. While the space dimension is determined by the multi-scale settings, the time dimension is typically set to half of the number of frames in the dynamic scene [14]. We set to examine if by using larger time dimension setting (and thus larger space-time planes) can help to further improve the visual quality of highly dynamic scenes.

For this evaluation, we use the "Longdress" sequence from the `8iVFB` dataset. We use the first 60 frames from this sequence, and we use two different time dimension settings: 30 and 60.



(a) PSNR (b) VMAF

Figure 3: RD-curve result for the "Longdress" sequence when using different time dimension.

Table 4: BD-Rate⇓, BD-PSNR⇑, BD-VMAF⇑ results on time dimension, using time dimension=30 as the anchor. Results show that the improvement is limited.

Settings	BD-Rate$_p$⇓	BD-PSNR⇑	BD-Rate$_v$⇓	BD-VMAF⇑
time_dim= 30	0%	0	0%	0
time_dim= 60	-4.0%	0.12	-4.4%	1.18

The RD-curve results for PSNR and VMAF are shown in Figure 3. The "rate" in the figure is varied by using different feature lengths $F \in \{4, 8, 16, 32\}$. The figure shows that these two different time dimension settings are comparable when compressing the "Longdress" sequence. At lower bitrates, i.e., shorter feature lengths, setting the time dimension to 30 gives better results; while at higher bitrates, using larger time dimension helps. However, the improvement is very limited. Table 4 shows the BD-Rate, BD-PSNR, and BD-VMAF results. These results are obtained using time dimension 30 as the anchor. Results show that by using the longer time dimension, the bitrate can be reduced by approximately 4% while achieving the same visual quality, and that the PSNR can improve by 0.12 dB with the same bitrate.

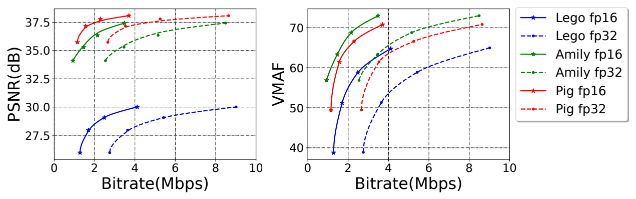
Given that "Longdress" is among the sequences with the most motion in our datasets, and that the improvement by using longer time dimension is very limited, we choose to use shorter time dimension (e.g., half of the number of frames) in the remaining experiments.

4.3 Model Precision

Mixed precision. PyTorch provides an automatic mixed precision package called TORCH.AMP [8]. It allows operations to use a mixture of `float32` and `float16` precision. This allows us to explore the feasibility of compacting the model by saving the model in `float16` and load the parameters later for inference. We note that `K-Planes` also use `float16` to speed up model training.

Further model compression. We explore lossless data compression via the `zip` tool [2] for further reducing the stored `K-Planes` representation size. While the trained models vary for different parameter settings and content, we observe that lossless data compression can further reduce the saved model size by 25% to 54%.

We conduct the experiment using all 11 sequences from all three datasets. For `8iVFB` and `Blender` datasets with 60 frames, we set the time dimension to 30. For `vsenseVVDB2` with 300 frames, we set the time dimension to 75. The trained models are further losslessly compressed via `zip`, and we use the `.zip` file size for calculating the "rate" for the rate-distortion curve. Figures 4 shows the RD-curve



(a) PSNR (b) VMAF

Figure 4: RD-curve result for the `Blender` dataset: `float16` vs. `float32`.

Table 5: BD-Rate⇓, BD-PSNR⇑, BD-VMAF⇑ results of `float32` on the `8iVFB` dataset, using `float16` as the anchor. (When using `float16` as the anchor, results of `float16` will be all 0s and are thus omitted in the table. Negative BD-PSNR and BD-VMAF results indicate that with the same model size, the quality of views rendered from `float16` models is better compared to `float32` models.)

Sequence	BD-Rate$_p$⇓	BD-PSNR⇑	BD-Rate$_v$⇓	BD-VMAF⇑
Longdress	120.3%	-1.67	120.2%	-17.05
Loot	128.7%	-2.74	128.9%	-19.45
Soldier	166.7%	-1.98	166.2%	-5.73
Redandblack	119.5%	-2.08	119.3%	-18.99

Table 6: BD-Rate⇓, BD-PSNR⇑, BD-VMAF⇑ results of `float32` on the `vsenseVVDB2` dataset, using `float16` as the anchor.

Sequence	BD-Rate$_p$⇓	BD-PSNR⇑	BD-Rate$_v$⇓	BD-VMAF⇑
Rafa	170.9%	-2.15	171.3%	-7.34
Lubna	163.6%	-2.75	162.8%	-10.44
Matis	132.7%	-1.99	132.4%	-15.15
Axeguy	174.1%	-1.82	174.6%	-7.44

Table 7: BD-Rate⇓, BD-PSNR⇑, BD-VMAF⇑ results of `float32` on the `Blender` dataset, using `float16` as the anchor.

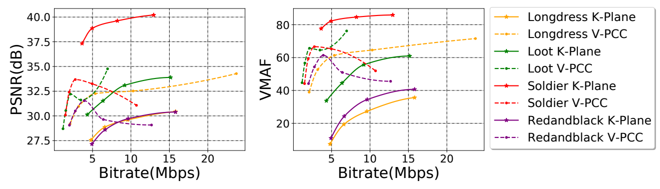
Sequence	BD-Rate$_p$⇓	BD-PSNR⇑	BD-Rate$_v$⇓	BD-VMAF⇑
Lego	115.7%	-2.28	114.7%	-14.92
Amily	142.6%	-2.35	143.3%	-11.47
Pig	128.9%	-1.39	128.7%	-12.45

results for both PSNR and VMAF for the `Blender` datasets. The BD-Rate, BD-PSNR, and BD-VMAF results of the three datasets are shown in Tables 5, 6, and 7. In these tables, the `float16` results are used as an anchor for calculating the BD-* results of using float32 for storing the trained model.

The RD-curve results show that the visual quality results of `float16` are comparable to `float32` while `float16` saves more than half of the saved model size. The BD-PSNR results show that when using the same bitrate for representing the dynamic scene, the visual quality of `float32` is 1.39 dB to 2.75 dB worse than `float16`, and the BD-VMAF results show that the VMAF results of `float32` is 5.73 to 19.45 worse than `float16`. Thus, we will use `float16` results for `K-Planes` vs. `V-PCC` comparison in the next section.

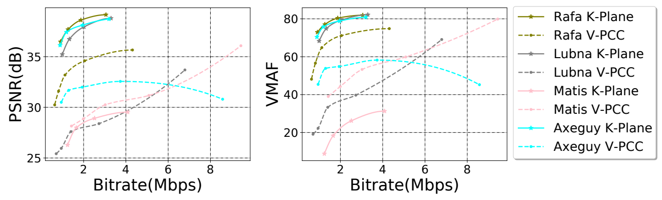
5 V-PCC VS. K-PLANES

In this section, we report our findings comparing `V-PCC` with `K-Planes` for dynamic 6-DoF volumetric video representation. For `V-PCC`, we



(a) PSNR (b) VMAF

Figure 5: RD-curve result for the `8iVFB` dataset.



(a) PSNR (b) VMAF

Figure 6: RD-curve result for the `vsenseVVDB2` dataset.

compress the raw `.ply` files using five different quantization parameter settings in Table 1. We obtain the `V-PCC` compressed binary file sizes and use them for calculating the "rate" in the RD-curve. We then decode and reconstruct the point cloud `.ply` files, render them, and compare them with the groundtruth test views. For `K-Planes` , we use multi-scale setting S=1,2, train the models for 150, 000 epochs with a learning rate of 0.001, and save the model in `float16`. We further losslessly compress the saved `K-Planes` models using the `.zip` tool and use the compressed `.zip` file for calculating the "rate". We do not use `zip` for compressed `V-PCC` binary files as no data deflation can be achieved.

5.1 Results of `8iVFB` and `vsenseVVDB2` Datasets

We discuss the results of the `8iVFB` and `vsenseVVDB2` datasets first since both datasets are carefully curated raw point cloud data and are made for `V-PCC` . The RD-curve results are shown in Figures 5 and 6. We notice that in a few cases, for `V-PCC`, with increased rate, e.g., the `r5` setting, the visual quality can become worse than lower rate, e.g., the `r3` setting. This finding is consistent with the subjective study performed by Cox et al. [12]. We have also checked our experiments and made sure it is the correct results. We thus report these results in the paper.

We report the BD-PSNR and BD-VMAF results in Tables 8 and 9. (We do not report the BD-Rate results in these tables because the RD-curves of comparative setups are very far away with no overlap in their "distortion" coordinates.) The results show that for the `8iVFB` dataset, `V-PCC` outperforms `K-Planes` in all but one ("Soldier") sequence; while for the `vsenseVVDB2` dataset, `K-Planes` outperforms `V-PCC` in all but one ("Matis").

We find that the performance of `K-Planes` and `V-PCC` appear to be correlated with the amount of motion in the dataset. For example, the four sequences that `K-Planes` perform well in (i.e., "Soldier", "Rafa", "Lubna", and "AxeGuy") are with little to moderate motion. For the remaining four highly dynamic sequences, however, `K-Planes` struggles to achieve a good performance, and `V-PCC` can compress them better.

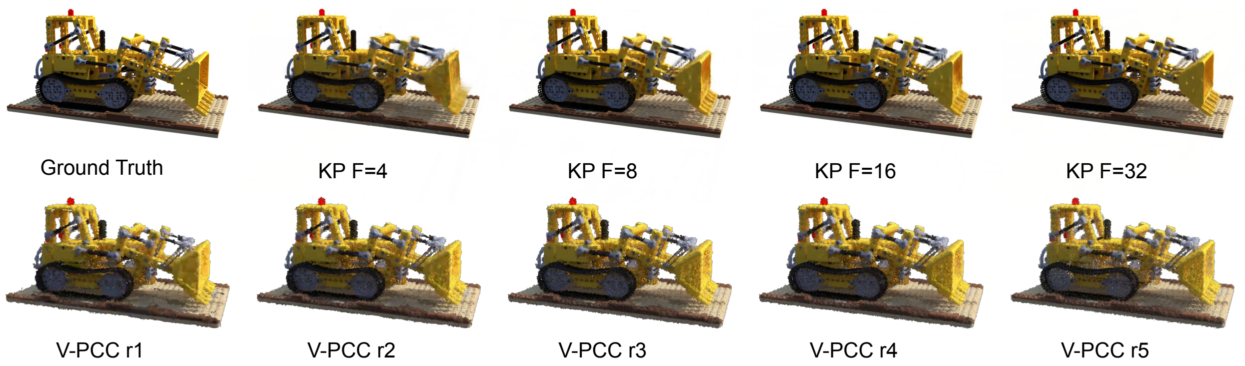


Figure 7: Visual quality comparison of "Lego" in the `Blender` dataset created by our team. The top row shows the groundtruth view and views rendered by trained `K-Planes` models with different feature length $F \in \{4, 8, 16, 32\}$. The bottom row shows views rendered by point clouds compressed by `V-PCC` in one of the five settings `r1` (lowest bitrate), r2, r3, r4, and `r5` (highest bitrate).

Table 8: `8iVFB` dataset: BD-PSNR⇑ and BD-VMAF⇑ results of `V-PCC` , using `K-Planes` as the anchor.

Sequence	BD-PSNR⇑	BD-VMAF⇑
Longdress	3.20	39.58
Loot	2.20	30.11
Soldier	-6.45	-21.29
Redandblack	0.49	21.20

Table 9: `vsenseVVDB2` dataset: BD-PSNR⇑ and BD-VMAF⇑ results of `V-PCC` , using `K-Planes` as the anchor.

Sequence	BD-PSNR⇑	BD-VMAF⇑
Rafa	-4.12	-10.10
Lubna	-9.69	-45.45
Matis	0.76	23.68
Axeguy	-6.06	-23.67

Table 10: `Blender` dataset: `V-PCC` results

Sequence	Metric	r1	r2	r3	r4	r5
Lego	PSNR (dB)	20.67	20.96	21.02	20.92	20.80
	VMAF	1.39	1.38	0.80	0.35	0.98
	Size (MB)	5.23	11.13	22.94	43.00	75.81
Amily	PSNR (dB)	26.55	26.29	25.63	24.97	24.92
	VMAF	8.65	5.50	0.95	0.01	0.01
	Size (MB)	0.84	1.31	2.86	8.25	20.21
Pig	PSNR (dB)	29.94	29.94	29.39	28.67	28.68
	VMAF	21.47	21.57	14.60	6.60	8.78
	Size (MB)	1.37	2.01	4.46	12.86	31.70

Table 11: `Blender` dataset: `K-Planes` results

Sequence	Metric	F=4	F=8	F=16	F=32
Lego	PSNR (dB)	25.97	27.96	29.07	30.01
	VMAF	38.78	51.19	58.78	64.76
	Size (MB)	1.28	1.70	2.48	4.11
Amily	PSNR (dB)	34.10	35.29	36.34	37.42
	VMAF	56.85	63.32	68.76	72.97
	Size (MB)	0.93	1.46	2.15	3.48
Pig	PSNR (dB)	35.73	37.16	37.79	38.08
	VMAF	49.34	61.45	66.56	70.78
	Size (MB)	1.15	1.56	2.29	3.70

5.2 Results of the `Blender` Dataset

For the three models in the `Blender` dataset, we first generate point clouds using a procedure that emulates real-world point cloud capture via RGB-Depth images. For this evaluation, we use the original textured mesh model for generating groundtruth test views.

We report our results in Tables 10 and 11. For `V-PCC`, while the structural information of the scene is correct, the visual quality is very low. For the "Lego" sequence, the PSNR is only about 20 dB; for "Amily", and "Pig", the PSNR results are lower than 30 dB. The obtained VMAF scores are also very low. We conjecture that the poor visual quality of `V-PCC` is partially caused by the point clouds recorded via RGB-Depth data, which is inherently noisy. In comparison, the `8iVFB` and `vsenseVVDB2` datasets are carefully curated. Additionally, another possible cause is that for `8iVFB` and `vsenseVVDB2` experiments, the groundtruth is rendered using the raw, uncompressed point cloud; while for the `Blender` experiments, the groundtruth is photorealistic rendering of the model. This results in significantly lower visual quality of `V-PCC`.

For `K-Planes`, the results are substantially better. For "Lego", the PSNR can be as high as more than 30 dB; while for "Amily" and "Pig", the PSNR can reach over 37 dB, with a `K-Planes` representation size of only about 4 MB (or 16 Mbps for the 2-second long sequence.) We further present visual results of four sets of rendered views of the "Lego" sequences in the `Blender` dataset in Figure 7.

6 CONCLUSION

In this paper, we performed a comparative study of a new dynamic neural scene representation model, `K-Planes`, and `V-PCC` for representing and efficiently transmitting 6-DoF volumetric video data. We find that for `K-Planes`, increasing the length of feature vectors can improve the visual quality faster than increasing the number of multi-scale planes. Results show that the current `K-Planes` models can outperform `V-PCC` when there is little to moderate amount of motion in the 6-DoF video sequence. We also find that in a volumetric data capturing scenario emulated by `Blender`, the visual quality of views rendered from `K-Planes` is significantly better than `V-PCC`.

ACKNOWLEDGMENTS

We appreciate constructive comments from anonymous referees. This work is partially supported by NSF under grants CNS-2200042 and CNS-2200048.

REFERENCES

[1] 2018. VMAF: The Journey Continues. https://medium.com/netflix-techblog/vmaf-the-journey-continues-44b51ee9ed12.

[2] 2019. zip - package and compress (archive) files. https://manpages.ubuntu.com/manpages/focal/man1/zip.1.html.

[3] 2020. Bjontegaard_metric. https://github.com/Anserw/Bjontegaard_metric.

[4] 2023. bjontegaard. https://pypi.org/project/bjontegaard/.

[5] 2023. Blender 3.5. https://www.blender.org/.

[6] 2023. Free | Amily Animations | Blender Demo. https://www.blender.org/download/demo-files/.

[7] 2023. Free | Piggy Animations | Vfx Grace. https://blendermarket.com/products/piggy-animations-vfx-grace.

[8] 2023. TORCH.AMP. https://pytorch.org/docs/stable/amp.html.

[9] 2024. Draco. https://google.github.io/draco/.

[10] Gisle Bjontegaard. 2001. Calculation of average PSNR differences between RD-curves. *VCEG-M33* (2001).

[11] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. 2022. TensoRF: Tensorial Radiance Fields. In *European Conference on Computer Vision (ECCV)*.

[12] Samuel Rhys Cox, May Lim, and Wei Tsang Ooi. 2023. VOLVQAD: An MPEG V-PCC Volumetric Video Quality Assessment Dataset. In *Proceedings of the 14th Conference on ACM Multimedia Systems*. 357–362.

[13] 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.

[14] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. 2023. K-Planes for Radiance Fields in Space, Time, and Appearance. arXiv:2301.10241 [cs.CV]

[15] 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.

[16] Dan Grois, Detlev Marpe, Amit Mulayoff, Benaya Itzhaky, and Ofer Hadar. 2013. Performance comparison of h. 265/mpeg-hevc, vp9, and h. 264/mpeg-avc encoders. In *2013 Picture Coding Symposium (PCS)*. IEEE, 394–397.

[17] Simon NB Gunkel, Rick Hindriks, Karim M El Assal, Hans M Stokking, Sylvie Dijkstra-Soudarissanane, Frank ter Haar, and Omar Niamut. 2021. VRComm: an end-to-end web system for real-time photorealistic social VR communication. In *Proceedings of the 12th ACM Multimedia Systems Conference*. 65–79.

[18] Bo Han, Yu Liu, and Feng Qian. 2020. ViVo: Visibility-aware mobile volumetric video streaming. In *Proceedings of the 26th annual international conference on mobile computing and networking*. 1–13.

[19] Henry Haugsten Hansen, Sayed Muchallil, Carsten Griwodz, Vetle Sillerud, and Fredrik Johanssen. 2020. Dense lidar point clouds from room-scale scans. In *Proceedings of the 11th ACM Multimedia Systems Conference*. 88–98.

[20] Branislav Jenco. 2022. *Virtual LiDAR error models in point cloud compression*. Master's thesis.

[21] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. 2022. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5521–5531.

[22] Rufael Mekuria, Kees Blom, and Pablo Cesar. 2016. Design, implementation, and evaluation of a point cloud codec for tele-immersive video. *IEEE Transactions on Circuits and Systems for Video Technology* 27, 4 (2016), 828–842.

[23] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

[24] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2021. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10318–10327.

[25] Sebastian Schwarz, Gaëlle Martin-Cocher, David Flynn, and Madhukar Budagavi. 2018. Common test conditions for point cloud compression. *Document ISO/IEC JTC1/SC29/WG11 w17766, Ljubljana, Slovenia* (2018).

[26] Sebastian Schwarz, Marius Preda, Vittorio Baroncini, Madhukar Budagavi, Pablo Cesar, Philip A Chou, Robert A Cohen, Maja Krivokuća, Sébastien Lasserre, Zhu Li, et al. 2018. Emerging MPEG standards for point cloud compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9, 1 (2018), 133–148.

[27] Shishir Subramanyam, Irene Viola, Jack Jansen, Evangelos Alexiou, Alan Hanjalic, and Pablo Cesar. 2022. Evaluating the impact of tiled user-adaptive real-time point cloud streaming on vr remote communication. In *Proceedings of the 30th ACM International Conference on Multimedia*. 3094–3103.

[28] Jeroen Van Der Hooft, Tim Wauters, Filip De Turck, Christian Timmerer, and Hermann Hellwagner. 2019. Towards 6dof http adaptive streaming through point cloud compression. In *Proceedings of the 27th ACM International Conference on Multimedia*. 2405–2413.

[29] Irene Viola, Jack Jansen, Shishir Subramanyam, Ignacio Reimat, and Pablo Cesar. 2023. VR2Gather: A Collaborative, Social Virtual Reality System for Adaptive, Multiparty Real-Time Communication. *IEEE MultiMedia* 30, 2 (2023), 48–59.

[30] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. 2003. Overview of the H. 264/AVC video coding standard. *IEEE Transactions on circuits and systems for video technology* 13, 7 (2003), 560–576.

[31] Emin Zerman, Cagri Ozcinar, Pan Gao, and Aljosa Smolic. 2020. Textured mesh vs coloured point cloud: A subjective study for volumetric video compression. In *Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*.