TEMPORAL CONDITIONAL CODING FOR DYNAMIC POINT CLOUD GEOMETRY COMPRESSION

Bowen Huang, Davi Lazzarotto and Touradj Ebrahimi

Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland

ABSTRACT

Point clouds allow for the representation of 3D multimedia content as a set of disconnected points in space. Their inherent irregular geometric nature poses a challenge to efficient compression, a critical operation for both storage and transmission. This paper proposes a VAE-inspired codec tailored for dynamic point cloud geometry compression, taking advantage of a temporal autoregressive hyperprior to enhance compression performance. Specifically, features derived from adjacent point cloud frames help build a hyperprior for conditional entropy coding. Sparse convolutions are leveraged to reach higher computational efficiency when compared to 3D dense convolutions. Remarkably, the proposed approach achieves an average 60.2% BD-rate gain against the contemporary V-PCC compression standard from MPEG.

***Index Terms*—** Point cloud compression, variational autoencoder, inter-frame coding

1. INTRODUCTION

As imaging modalities advance towards immersive representations, the creation of 3D multimedia content has been growing exponentially. Among the representation methods, point clouds have emerged as a significant modality to portray 3D multimedia signals in different key applications such as augmented/virtual reality and autonomous driving. Given the potentially vast amount of points within point clouds, the associated large data volumes present considerable obstacles for both storage and transmission. Different from 2D images and video, where pixels are distributed on uniform grids, point clouds contain points that can be irregularly sampled from an underlying surface. Moreover, the accuracy of attribute coding, such as colors, is tied to the performance of geometry compression. Thus, efficient point cloud geometry compression (PCGC) techniques are imperative.

Both handcrafted and learning-based approaches have been proposed for PCGC, although the majority are designed to code static point clouds. For 2D video coding, inter-frame prediction has been demonstrated to be effective. However, its transposition to dynamic PCGC is non-trivial. The misalignment of coordinates in successive frames, combined with the inefficiencies in motion vector estimation and transmission in 3D space, make the task of directly adapting video coding techniques for point clouds particularly challenging. To address these limitations, this paper proposes a variational autoencoder (VAE)-inspired coding framework that leverages the temporal relationship between frames as hyperpriors for the entropy model. The architecture is built using sparse convolutional layers, which are computationally more efficient than their dense counterparts. Conducted evaluations on the 8i dataset reveal that the proposed approach achieves a 60.2% BD-rate gain based on the point-to-point PSNR (D1 PSNR) metric over the handcrafted V-PCC in inter mode and surpasses learning-based PCGC methods restricted to static scenarios.

The authors would like to acknowledge support from the Swiss National Scientific Research project entitled "Compression of Visual information for Humans and Machines (CoViHM)" under grant number 200020_207918.

2. RELATED WORK

Several approaches have been proposed for PCGC. Two distinct alternatives have been selected for the MPEG compression standards in Geometry-based Point Cloud Compression (G-PCC) and in Video-based Point Cloud Compression (V-PCC), being useful in different scenarios [1]. In V-PCC, point clouds are mapped and structured into 2D frames which are encoded using conventional video codecs. G-PCC employs an octree to systematically partition the 3D space, denoting the occupancy of the sub-cubes through binary codes. While both standards have consistently been used for benchmarking the performance of the compression methods in the literature, their performance is limited by their manually crafted rules.

Drawing inspiration from the remarkable achievements of learning-based methods in image and video compression, similar architectures have been adopted for PCGC. For static PCGC, early works employed dense 3D convolutions in autoencoder architectures for lossy PCGC [2, 3, 4] and block prediction [5]. Alternatively, voxel occupancy values were directly estimated [6] for lossless coding approaches. Sparse convolutions have been later leveraged both for lossy [7, 8] and lossless [9] coding of point clouds, achieving better performance and reduced complexity. Other methods use point coordinates directly as input [10, 11], and octree is also used for the compression of point clouds with lower density such

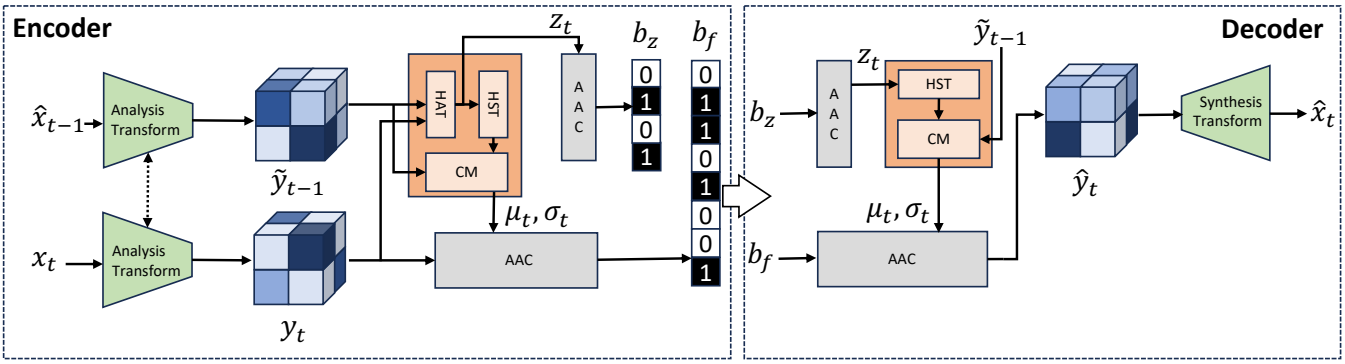


Fig. 1: Overview of the proposed method. The feature extractors for x_t and $\hat{x}_{t-1}$ share the same weights, and the extracted features $y_t, \tilde{y}_{t-1}$ are used to estimate the hyperprior z_t in the Hyperprior Analysis Transform (HAT). To get the conditional entropy coding parameters μ_t and σ_t, z_t is decoded by the Hyperprior Synthesis Transform (HST) and fused with $\tilde{y}_{t-1}$ in the Context Model (CM). At the decoder side, the same HST and CM are applied to the decoded z_t and $\tilde{y}_{t-1}$ to generate μ_t, σ_t for entropy decoding, and the decoded features are then reconstructed into $\hat{x}_t$.

as those derived from LiDAR scans [12, 13].

In the field of dynamic PCGC, Akhtar et al. [14] introduced a predictor based on sparse convolutions. This predictor exploits multi-scale features from the preceding frame to yield a prediction for the current frame, subsequently transmitting the residual for bitrate reduction. Notably, the absence of explicit motion estimation and motion compensation (MEMC) in Akhtar's architecture led Fan et al. [15] to conceive D-DPCC, which integrated an end-to-end feature-domain MEMC component with explicit motion vectors. This module was later augmented with multi-head attention mechanisms and multi-resolution MEMC in their successive work [16]. Nonetheless, since the entropy of the residual signal is greater than or equal to the conditional entropy between adjacent frames [17], effectively modeling the conditional probability distribution can theoretically allow for a different solution from explicit MEMC for dynamic PCGC.

3. METHODOLOGY

3.1. Rate-distortion autoencoder

The architecture of the proposed dynamic PCGC network is inspired by the VAE framework based on sparse convolutional layers, with a temporal autoregressive hyperprior and a context coding model being added to remove temporal redundancies during entropy coding. Within the VAE framework, the encoder condensates input information into latent variables, while the decoder reconstructs a faithful representation of the point cloud from the transmitted latent variables. Given an input point cloud x, the optimization target of the network can be represented as a rate-distortion optimization problem given by Equation 1, where λ is the Lagrange multiplier that determines the rate-distortion trade-off, $f(\cdot)$ and $g(\cdot)$ are the encoder and decoder respectively, and quantization is symbolized by $\lfloor \cdot \rceil$.

$$\mathbb{E}_{x\sim p_x}[-\log p_y(\lfloor f(x) \rceil)] + \lambda \cdot \mathbb{E}_{x\sim p_x}[d(x, g(\lfloor f(x) \rceil)] \quad (1)$$

In the proposed framework, the static PCGC method PCGCv2 [7] is employed for individual point cloud frames. The analysis transform is designed with three sequentially linked Downsampling Blocks, downsampling the input point cloud frame by a factor of 8, and hierarchically aggregating spatial features. In tandem, the feature synthesis transform is composed of three sequentially linked Upsampling Blocks, each capable of estimating the occupancy likelihood for voxels. Similarly to PCGCv2, adaptive pruning is applied at each upsampling stage, retaining only the N_k voxels with the highest occupancy probability at each stage k, with N_k provided by the analysis transform.

3.2. Temporal autoregressive hyperprior and context model

For each point cloud frame, the extracted latent features are modeled as Gaussian random variables convolved with a unit uniform distribution. For the latent variables y_t generated from a point cloud frame x_t at a time t, its mean μ_t and scale σ_t are predicted conditioned on both the temporal autoregressive hyperprior z_t and the latent features $\tilde{y}_{t-1}$ generated from the previous decoded frame $\hat{x}_{t-1}$, as depicted in Fig. 2. To leverage the temporal redundancy, z_t is inferred from both the present and the previous frame through the hyperanalysis transform, improving the entropy coding efficiency by embedding temporal dependencies directly in the hyperprior.

To enhance the fidelity of the estimated parameters μ_t, σ_t, latent variables $\tilde{y}_{t-1}$ obtained from the previous decoded frame are also combined with the decoded hyperprior $\hat{z}_t$ in the context model, as shown in Fig. 2. These concatenated tensors undergo downsampling to facilitate integration

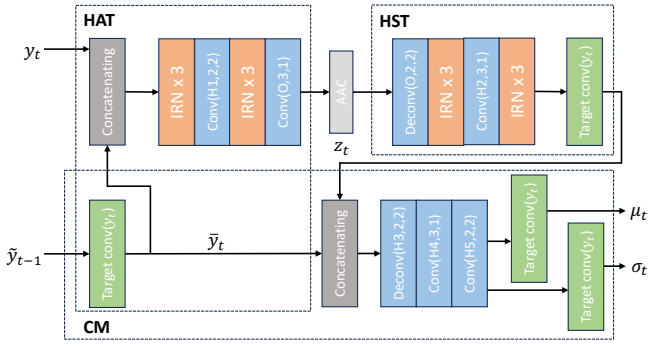


Fig. 2: The detailed architecture of the temporal autoregressive hyperprior estimation module combined with CM. Each convolution layer is represented by Conv(n, m, s), with n being the number of filters, m the kernel size and s the stride. Similarly, Deconv(n, m, s) layers are used for upsampling. Each layer is followed by a ReLU layer for non-linear activation and the Inception-Residual Block Net (IRN)[18] is used for local feature analysis and aggregation. Within the context of each target convolution layer, the output tensor shares the same coordinates as y_t.

with the decoded hyperprior across various scales and target convolution[14] is used at different points of the network to align the coordinates of the tensors to those of the latent features y_t. Subsequently, the Gaussian parameters μ_t, σ_t are generated as the output of the context model for entropy coding of the latent variables y_t.

Compared to methods relying on MEMC, the proposed temporal autoregressive hyperprior and context model exploit temporal dependencies without the computation of motion vectors and residuals, circumventing the need to encode this information within the bitstream.

3.3. Loss function and coding procedure

As described in Eq. 1, the loss function can be modeled as a joint optimization problem of rate and distortion $R + \lambda \cdot D$. Since z_t serves as side information, it must be counted in the bitstream as well. Therefore, the rate can be estimated according to Equation 2, while the distortion is calculated by the binary cross entropy (BCE) loss following PCGCv2, as given by Equation 3.

$$\mathcal{R} = \mathbb{E}_{x\sim p_x}[\sum_t (\log p(y_t|y_{t-1}, z_t) + \log p(z_t))] \quad (2)$$

$$\mathcal{D} = \frac{1}{K}\sum_k (\frac{1}{N_k}\sum_v -(\mathcal{O}_v \log p_v + (1-\mathcal{O}_v)\log(1-p_v))) \quad (3)$$

In Equation 3, $\mathcal{O}_v$ is the ground truth occupancy value for the voxel v, and N_k is the number of points of the upsampling stage k. To minimize the distortion, the BCEs sourced from all K stages within the synthesis transform are aggregated through the average operation, yielding the final distortion measure.

The overall architecture of the framework is illustrated in Fig. 1. For encoding a point cloud frame x_t, the coding frame and its reconstructed predecessor $\hat{x}_{t-1}$ undergo the analysis transform. Subsequently, the latent features from the previous frame $\tilde{y}_{t-1}$ are fed to a convolution layer targeted at the coordinates of y_t. The resulting variables are then concatenated, forming the basis for the estimation of the hyperprior z_t, which is encoded using a factorized entropy coder and added to the bitstream as side information. This side bitstream is then decoded back to z_t and combined with $\tilde{y}_{t-1}$ in the context model to generate the parameters for the Gaussian coding model. The synthesis transform finally takes as input these decoded features to produce the final output $\hat{x}_t$. Since the entropy coding module is able to only encode the features of the sparse tensor y_t, the coordinates of the latent variables are encoded and decoded by G-PCC, contributing only to a small fraction of the total bitrate.

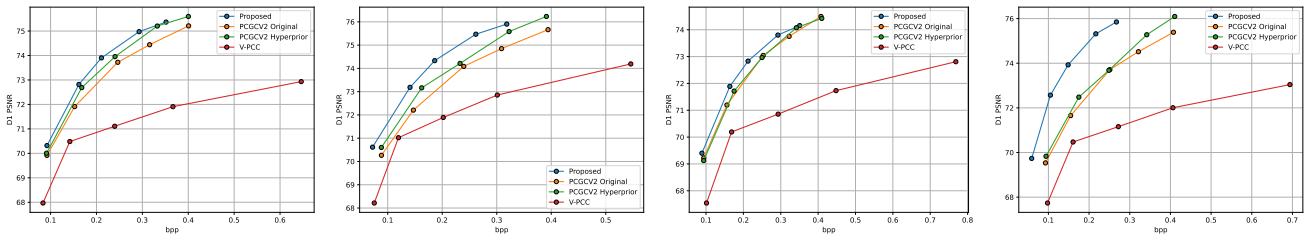
The input sequence of point cloud frames is partitioned into frame groups of constant size. While the majority of the frames are encoded following the process illustrated in Fig. 1, previous frames are not available for the estimation of the mean and scale of the latent features relative to the first frame of the group. In that case, a modified version of PCGCv2 is used for intra-coding, where the hyperprior is derived solely from y_t, being denominated PCGCv2-hyper, which is separately trained with the same strategy.

4. EXPERIMENTS

4.1. Experimental settings

The training dataset is derived from the UVG-VPC dataset [19]. Specifically, this dataset encompasses 12 sequences with 3000 frames. Each point cloud frame is previously downsampled to have a spatial resolution confined to 9 bits. For performance assessment, the 8iVFB dataset [20] is used. Adhering to the MPEG common test condition (CTC), only the initial 100 frames from each sequence are used. The bitrate is represented in terms of bits per point (bpp), while the quality of reconstruction is quantified using the point-to-point geometry (D1) PSNR.

The proposed network is trained with $\lambda = 1, 2, 3, 5, 7$ for 50 epochs with a batch size equal to 4. The Adam optimizer with $\beta = (0.9, 0.999)$, an initial learning rate of 0.008, and a scheduler with a decay rate of 0.7 for every 15 epochs are used. To accelerate the convergence, the λ is set to 20 for the first 10 epochs and set to its original value for the rest 40 epochs. All the experiments are conducted on a GeForce RTX 3090 GPU with 24GB memory. The proposed method is compared with the state-of-the-art rule-based point cloud compression method V-PCC and static PCGC method PCGCv2.



(a) *Longdress* (b) *Loot* (c) *Redandblack* (d) *Soldier*

Fig. 3: D1 Rate-Distortion curves on 8iVFB test sequences. For the proposed approach, the frame group size is determined by the sequence's length. In contrast, V-PCC adopts a frame group size of 32.

For the sake of equitable comparison, the PCGCv2 is evaluated utilizing the pre-trained checkpoints provided by the author. Additionally, a comparison to the same network used for intra-coding, i.e. PCGCv2-hyper, is also conducted. Finally, results extracted from the papers [14, 15] were also used for comparison to the proposed method.

Table 1: BD-Rate (%) gains of the proposed method against V-PCC, PCGCv2, PCGCv2-hyper, and the proposed network with different frame group size. GoF-20 and GoF-4 indicate frame group sizes of 20 and 4 respectively.

Methods	Longdress	Loot	Redandblack	Soldier	**Average**
V-PCC	-59.72	-52.49	-56.72	-71.44	**-60.17**
PCGCv2	-16.27	-26.72	-11.09	-44.32	**-24.60**
PCGCv2-hyper	-6.78	-16.17	-10.05	-39.90	**-18.22**
GoF-20	-0.18	-1.68	-0.22	-2.45	**-1.13**
GoF-4	-1.66	-4.98	-2.43	-13.34	**-5.60**

Table 2: BD-Rate (%) gains of other methods against V-PCC.

Methods	Redandblack	Soldier
Akhtar et al. [14]	-55.58	-43.60
Fan et al. [15]	-68.97	-85.71

4.2. Performance comparison

The rate-distortion plots for the evaluated methods are depicted in Fig. 3, with the corresponding BD-Rate gains enumerated in Tab. 1. Contrasting the projection-based methodology V-PCC, the proposed method natively targets 3D space compression with powerful learning-based modules and culminates in BD-Rate reduction exceeding 60.2% on average. Furthermore, it achieves an average BD-Rate reduction relative to PCGCv2 surpassing 24.6%. For sequences with small motion amplitude such as Soldier and Loot, the temporal correlation is stronger, and the improvement is even more pronounced. These results show that the incorporation of the temporal autoregressive hyperprior combined with context modeling is capable of leveraging inter-frame correlations to reduce bitrate. It's worth noting that with only the spatial hyperprior in PCGCv2-hyper, there are still BD-Rate gains against the original PCGCv2.

For other learning-based methods for dynamic PCGC [14, 15], the BD-Rate gains against V-PCC on the two sequences of the 8iVFB dataset reported in the original papers are included in Tab. 2. It can be observed that the proposed method offers a larger rate reduction against V-PCC when compared to [14]. However, if the performance of the proposed method is lower than [15], it requires a lower memory footprint: the number of parameters is 1.327M when compared to the 3.017M parameters of [15]. Moreover, the complexity is also reduced, with an average encoding and decoding time per frame of 0.45 and 0.38 seconds respectively, when compared to the values of 1.2 and 1.09 obtained for [15] in the same platform.

Tab.1 also provides insights into the outcomes derived from varying frame group sizes. Experimental outcomes substantiate that as the frame group size increases from 4 and 20 to 100 (spanning the entire sequence), the compression performance is enhanced as an augmented number of frames are processed in the inter-frame mode.

5. CONCLUSION

This work presented a VAE-based geometric compression framework for dynamic point clouds, leveraging a temporal autoregressive hyperprior associated with context modeling to exploit temporal redundancies for bitrate reduction. Latent features derived from previous frames are used to produce the downsampled hyperprior that is encoded in the bitstream as side information, serving as input to the context model in their original resolution to improve the accuracy of the generated parameters for the Gaussian coding model. Empirical assessments underscore the efficacy of the proposed approach, registering a 60.2% BD-Rate reduction over the handcrafted V-PCC codec and a 24.6% BD-Rate reduction relative to the static PCGC methodology PCGCv2. Future work may focus on improving the compression performance by utilizing novel network architectures for better feature analysis, as well as incorporating low-complexity MEMC techniques.

6. REFERENCES

[1] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai, "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*, vol. 9, pp. e13, 2020, Publisher: Cambridge University Press.

[2] Maurice Quach, Giuseppe Valenzise, and Frederic Dufaux, "Learning convolutional transforms for lossy point cloud geometry compression," in *2019 IEEE international conference on image processing (ICIP)*. IEEE, 2019, pp. 4320–4324.

[3] Davi Lazzarotto and Touradj Ebrahimi, "Learning residual coding for point clouds," in *Applications of Digital Image Processing XLIV*. SPIE, 2021, vol. 11842, pp. 223–235.

[4] Nicolas Frank, Davi Lazzarotto, and Touradj Ebrahimi, "Latent space slicing for enhanced entropy modeling in learning-based point cloud geometry compression," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 4878–4882.

[5] Davi Lazzarotto, Evangelos Alexiou, and Touradj Ebrahimi, "On block prediction for learning-based point cloud compression," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 3378–3382.

[6] Dat Thanh Nguyen, Maurice Quach, Giuseppe Valenzise, and Pierre Duhamel, "Learning-Based Lossless Compression of 3D Point Cloud Geometry," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, June 2021, pp. 4220–4224, ISSN: 2379-190X.

[7] Jianqiang Wang, Dandan Ding, Zhu Li, and Zhan Ma, "Multiscale Point Cloud Geometry Compression," Nov. 2020, arXiv:2011.03799 [cs, eess].

[8] Davi Lazzarotto and Touradj Ebrahimi, "Evaluating the effect of sparse convolutions on point cloud compression," in *11th European Workshop on Visual Information Processing (EUVIP)*, 2023.

[9] Dat Thanh Nguyen and André Kaup, "Learning-based lossless point cloud geometry coding using sparse tensors," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 2341–2345.

[10] Wei Yan, Yiting shao, Shan Liu, Thomas H. Li, Zhu Li, and Ge Li, "Deep AutoEncoder-based Lossy Geometry Compression for Point Clouds," Apr. 2019, arXiv:1905.03691 [cs, eess].

[11] Tianxin Huang and Yong Liu, "3D Point Cloud Geometry Compression on Deep Learning," in *Proceedings of the 27th ACM International Conference on Multimedia*, New York, NY, USA, Oct. 2019, MM '19, pp. 890–898, Association for Computing Machinery.

[12] Lila Huang, Shenlong Wang, Kelvin Wong, Jerry Liu, and Raquel Urtasun, "OctSqueeze: Octree-Structured Entropy Model for LiDAR Compression," Jan. 2021, arXiv:2005.07178 [cs, eess].

[13] Chunyang Fu, Ge Li, Rui Song, Wei Gao, and Shan Liu, "OctAttention: Octree-Based Large-Scale Contexts Model for Point Cloud Compression," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 1, pp. 625–633, June 2022, arXiv:2202.06028 [cs].

[14] Anique Akhtar, Zhu Li, and Geert Van der Auwera, "Inter-Frame Compression for Dynamic Point Cloud Geometry Coding," July 2022, arXiv:2207.12554 [cs, eess].

[15] Tingyu Fan, Linyao Gao, Yiling Xu, Zhu Li, and Dong Wang, "D-DPCC: Deep Dynamic Point Cloud Compression via 3D Motion Prediction," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, Vienna, Austria, July 2022, pp. 898–904, International Joint Conferences on Artificial Intelligence Organization.

[16] Shuting Xia, Tingyu Fan, Yiling Xu, Jenq-Neng Hwang, and Zhu Li, "Learning Dynamic Point Cloud Compression via Hierarchical Inter-frame Block Matching," May 2023, arXiv:2305.05356 [cs, eess].

[17] Yung-Han Ho, Chih-Peng Chang, Peng-Yu Chen, Alessandro Gnutti, and Wen-Hsiao Peng, "Canf-vc: Conditional augmented normalizing flows for video compression," in *European Conference on Computer Vision*. Springer, 2022, pp. 207–223.

[18] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. 2017, AAAI'17, p. 4278–4284, AAAI Press.

[19] Guillaume Gautier, Alexandre Mercat, Louis Fréneau, Mikko Pitkänen, and Jarno Vanne, "UVG-VPC: Voxelized Point Cloud Dataset for Visual Volumetric Video-based Coding," in *2023 15th International Conference on Quality of Multimedia Experience (QoMEX)*, June 2023, pp. 244–247, ISSN: 2472-7814.

[20] d'Eon Eugene, Harrison Bob, Myers Taos, and A Chou Philip, "8i voxelized full bodies-a voxelized point cloud dataset," 2017.