

Using Fractional Super-Resolution to Improve Lossy Compression of Point Cloud Geometry

Tomás M. Borges, Renan U. B. Ferreira, Diogo C. Garcia, Ricardo L. de Queiroz

Abstract—We present a method for post-processing point clouds’ geometric information by applying a previously proposed fractional super-resolution technique to clouds encoded and decoded with MPEG’s G-PCC codec. In some sense, this is a continuation of that previous work, which requires only a down-scaled point cloud and a scaling factor, both of which are provided by the G-PCC codec. For non-solid point clouds, an *a priori* down-scaling is required for improved efficiency. The method is compared to the G-PCC itself, as well as machine-learning-based techniques. Results show a great improvement in quality over G-PCC and comparable performance to the latter techniques, with the advantage of not needing any kind of previous training.

Index Terms—Point cloud compression, point cloud processing, G-PCC, super-resolution

I. INTRODUCTION

Point clouds (PC) are sparse 3D signals composed by geometry and attributes information (which may include color, reflectance, normal vectors, etc.), and have been in the spotlight of researchers in recent years for its usability in applications such as augmented and virtual reality (AR/VR) [1], telecommunications [2], autonomous vehicle [3] and world heritage [4]. Because of the need for compression for either storage or transmission, the Motion Picture Experts Group (MPEG) has been directing efforts for the compression of PCs in two fronts, i.e., geometry-based PC compression (G-PCC) and video-based PC compression (V-PCC). The latter uses video codecs for encoding its projection in a plane, while the former uses the *octree* structure [5].

The geometric information of a PC may be expressed as a list V of unordered ternary coordinates, such that the n -th point is $\mathbf{v}_n = (x_n, y_n, z_n)$. To encode such information, G-PCC proposes to represent V as an octree, the quality of the encoded geometry can be tuned by pruning the octree using a down-scale process. This is performed together with a coordinate transformation at the encoder, such that, for the n -th point of V_d

$$\mathbf{v}_{d_n} = \text{round} \left(\frac{\mathbf{v}_n - T}{s} \right), \quad (1)$$

Work partially supported by CNPq under grants 88887.600000/2021-00 and 301647/2018-6.

T. M. Borges is with the Electrical Engineering Department at University of Brasilia, Brazil, e-mail: tomas@divp.org, ORCID: 0000-0003-0182-1866.

R. U. B. Ferreira is with the Gama Engineering College, University of Brasilia, Brazil, e-mail: renan@unb.br, ORCID: 0000-0003-4324-5238.

D. Garcia is with the Gama Engineering College, University of Brasilia, Brazil, e-mail: diogogarcia@unb.br, ORCID: 0000-0002-3816-0873.

R. de Queiroz is with the Department of Computer Science, University of Brasilia, Brazil, e-mail: queiroz@ieec.org, ORCID: 0000-0002-3911-1838.

Digital Object Identifier: 10.14209/jcis.2023.19

where, $\text{round}(\cdot)$ is the function that rounds the components of a vector to the nearest integer,

$$T = (\min x_n, \min y_n, \min z_n), \quad (2)$$

and $s > 1$ is the scale factor [6]. This scaling reduces the number of points to be encoded due to rounding and to duplicate point removal, making V_d a coarser geometry when compared to V . The larger the scale factor s , the lower the encoding rate, and consequently, the lower the number of output points, and the coarser the geometry. At the decoder side, the scaled geometry is expanded and shifted back to its original position using the same values of s and T . Fig. 1 exemplifies the down-scale for $s = 2$ and $s = 4$.

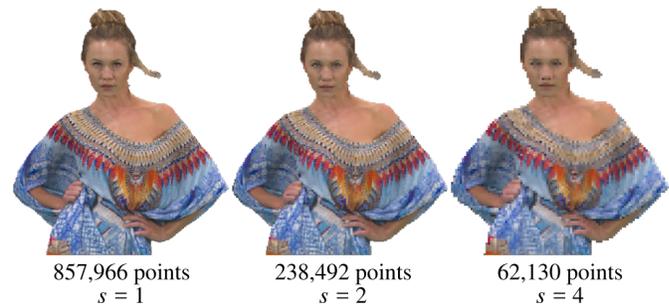


Fig. 1. Example of the down-scale used in G-PCC.

Since only expansion is performed at the decoder, i.e.,

$$\mathbf{v}_{dec_n} = \text{round}(\mathbf{v}_{d_n} \cdot s) + T, \quad (3)$$

the down-scaled voxels become quickly sparse with the increasing of s . The empty space between points is usually filled by rendering bigger voxels, giving a blocky aspect to the PC.

Some techniques have been proposed in order to improve coding efficiency of the geometry for these lossy cases, such as the use of slicing interpolation [7] inside G-PCC or even completely new approaches, such as using dyadic decomposition [8]. Moreover, the use of a lookup table (LUT) based on neighborhood inheritance has been used for context generation on geometry coding of dynamic voxelized PCs [9] as a replacement to the octree, providing extra control for lossy-geometry compression.

Another approach to improve lossy-geometry compression is to perform some form of interpolation with the down-scaled voxels. Although there are many interpolation or super-resolution (SR) methods for PC geometry [10], [11], [12], [13], [14], [15], [16], [17], they are not well-suited for the octree structure used in G-PCC. Recently, Borges *et al.* [18] proposed the use of LUTs relating the downsampled neighbourhood of

a given voxel with its children occupancy to super-resolve voxelized PCs downsampled at arbitrary fractional scales. In this paper, we propose, as a continuation of that work, to apply this SR technique as a post-processing tool for PCs encoded/decoded with G-PCC, and compare the results with some learning-based techniques for PCC.

II. POST-PROCESSING METHOD

In a nutshell, the SR technique from Borges *et al.* [18] (“frac SR”) works by first downsampling once more the input geometry (V_d) using the same value of s , creating V_{d^2} . Relating the neighborhood configurations from V_{d^2} with the occupancy of child nodes from V_d , they create a LUT. Fig. 2 illustrates the mapping of neighborhood and child occupancies, and the built LUT. Assuming that self-similarities are somewhat maintained at different scales, and taking into account the irregularities of fractional downsampling, one can query the built LUT with the neighborhoods from V_d to find its children occupancy, thus super-resolving V_d into V_{sr} .

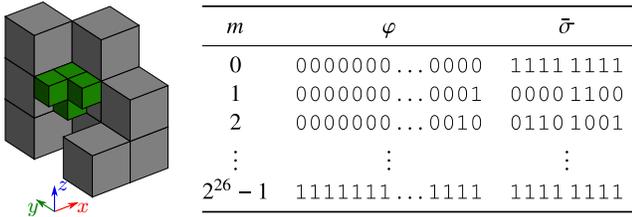


Fig. 2. On the left, a neighbourhood (φ) from V_{d^2} and a its related child occupancy (σ) from V_d . On the right, the built LUT, $\bar{\sigma}(m)$ is the bitwise mean of all child occupancies sharing the same neighborhood configuration.

Since this technique can be employed for arbitrary scale factors, it can be used to improve quality of PCs downsampled using Eq. (1), provided that s is known. Thus, it can be used to post-process decoded outputs from the G-PCC codec in lossy-geometry configuration.

As reported by the authors, better results are found when $1 < s \leq 2$. For coding conditions where $s > 2$, “frac SR” may be applied successively. Fig. 3 shows the general idea of the proposed post-processing method.

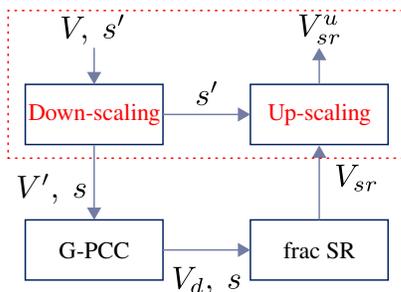


Fig. 3. The fractional SR is applied to the decoded PC using the same scale factor as used in the downsampling of the G-PCC codec. For sparse clouds, a down-scaling is applied before G-PCC encoding, and then an up-scaling is applied after the SR process.

The SR method relies on the use of the adjacent neighbourhood of a given voxel in order to predict its child nodes. For some sparser PCs, the immediate neighbours may not be

available for some (or maybe all) voxels. In order to surpass this problem, we down-scale the input PC by a factor s' prior to the encoding. Then, a subsequent up-scaling of the cloud V_{sr} by the same factor is necessary, after the “frac SR” step. When dealing with solid PCs, we may set $s' = 1$, and the diagram of Fig. 3 remains valid.

III. EXPERIMENTAL RESULTS

To test the post-processing method, we propose two distinct experiments, according to the sparsity of the input PCs, for solid (voxelized PCs with continuous surface) and non-solid clouds, according to the categories used in MPEG [19]. In the first one, we apply the fractional SR directly to the decoded versions of solid PCs ($s' = 1$ in Fig. 3). The scaling factor is defined by MPEG G-PCC’s Common Test Conditions (CTC) [20] according to the PC’s geometry for each rate point to be tested. The greater the rate id (as in “R6”), the greater the bit rate, the closer to 1 is the value of s .

In the second experiment, for testing non-solid PCs, we apply a down-scaling prior to encoding. The scaling factor s' is defined as the highest power of 2 that makes V_d keep approximately the same number of points as V , i.e., we enforce a down-scaling only to *densify* the input geometry, avoiding over-decimating its points. According to the CTC’s lossy-geometry section, the largest possible value used for this scaling would be 2048 (defined for PCs with 20 bits in geometry precision). This means we only need 4 bits to transmit s' as side information. Since the bitrate is calculated in bits per input points, i.e., the number of points in the original PC (which vary from hundreds of thousands to millions), these 4 bits are negligible. In Tab. I, we summarize the information of the tested PCs.

TABLE I
TESTED CLOUDS AND THEIR CORRESPONDENT DOWN-SCALING FACTOR.

Point cloud	s'
dancer_vox11_00000001 [21]	1
longdress_vox10_1300 [22]	1
loot_vox10_1200 [22]	1
queen_0200	1
redandblack_vox10_1550	1
soldier_vox10_0690	1
house_without_roof_00057_vox12	2
statue_klimt_vox12	4

For both experiments, we compare our results to those of G-PCC as well as to the following machine learning (ML) based end-to-end compression techniques: PCC-GEO-CNN-v2 [23], PCGCv2 [24] and ADL-PCC [25]. Those methods were recently the object of a study for AI-based solutions for PCC performed by Zaghetto [26].

PCC-GEO-CNN-v2, by Quach *et al.* [23], is focused on lossy compression of static PC geometry using deep convolution networks (CNN). It proposes improvements over a previous version [27], using a scale hyperprior model for entropy coding, deeper transforms, a different balancing weight in the focal loss, optimal thresholding for decoding and sequential model training. PCGCv2, by Wang *et al.* [24], proposes a geometry compression framework utilizing sparse convolution.

Their framework includes both lossless and lossy geometry compression, and it also can provide scalable coding capability. The method represents the PC using a sparse tensor and employs spatially sparse CNNs for processing. Specifically, the sparse CNNs are employed to exploit the spatial dependency between voxels and predict the occupancy probability, which are used for entropy coding or binary classification of voxel occupancy symbols. Finally, ADL-PCC, by Guarda *et al.* [25], proposes to use multiple deep learning coding models that are selectively applied to encode individual blocks of the PC, enabling efficient adaptation to different content characteristics. Key components include block-based encoding, an auto-encoder for generating latent representations, discretization for entropy coding, and the use of a variational autoencoder to estimate entropy model parameters. The method also employs a deep-learning coding model selection based on reconstruction quality and rate evaluation, and reconstructs the full PC through merging the encoded blocks.

It is important to note we only bring the results for the sequences used in the ML tests, for a direct comparison. Some of the clouds in G-PCC's CTC were used in the training of some of these models, and thus could not be used for testing.

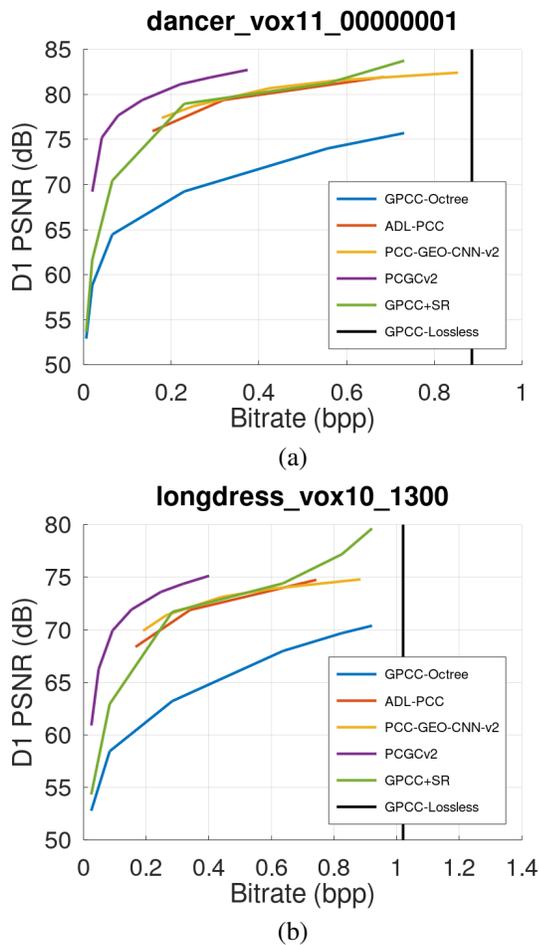


Fig. 4. Results under D1 PSNR metric for solid PCs: (a) *dancer*, (b) *longdress*.

Figs. 4 and 5 show the rate-distortion curves for some of the obtained results, for solid and non-solid PCs, respectively. The rate is measured in bits per input points (number of voxels in

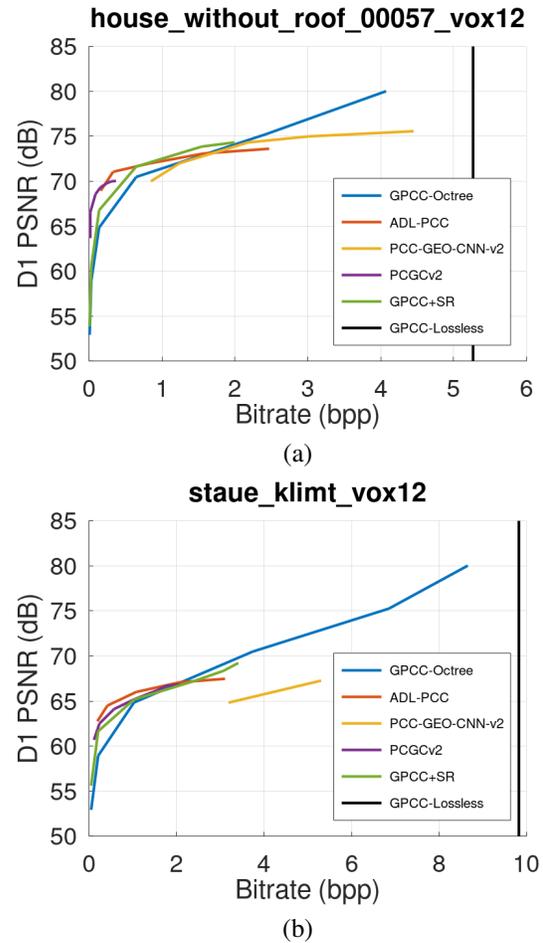


Fig. 5. Results under D1 PSNR metric for non-solid PCs:(a) *house_without_roof*, (b) *stau_klimt*.

the input PC), while the distortion is measured with the point-to-point PSNR (D1 PSNR) metric [28].

We note that our results bring great improvement over G-PCC, mainly for higher rates. We also note that they are comparable to those of ML-based techniques, except for PCGCv2, which outperforms the others, particularly for solid clouds. One should consider, however, that all presented ML techniques require previous training of tuned models *for each* intended bitrate, while the proposed method is simpler and easily adaptable for any given rate.

We also notice a drop in the lower rates region of the curves. At this region, the scale factor s is greater than 2, requiring to use the fractional SR more than once. This is not ideal because every time we apply it, we imply the neighbourhood for the next application, which may propagate errors.

In Tab. II we bring the BD-rate comparison of all results to that of G-PCC for solid and non-solid PCs, respectively. For the solid PCs, we also show a column comparing our results considering only the four higher rate points (“frac SR (HR)”), i.e. the four rightmost points from the graphs in Fig. 4, where the rate ranges are closer to most of those from the ML techniques, bringing therefore a more adequate comparison. One can observe that non-solid PCs (Fig. 5) are harder to

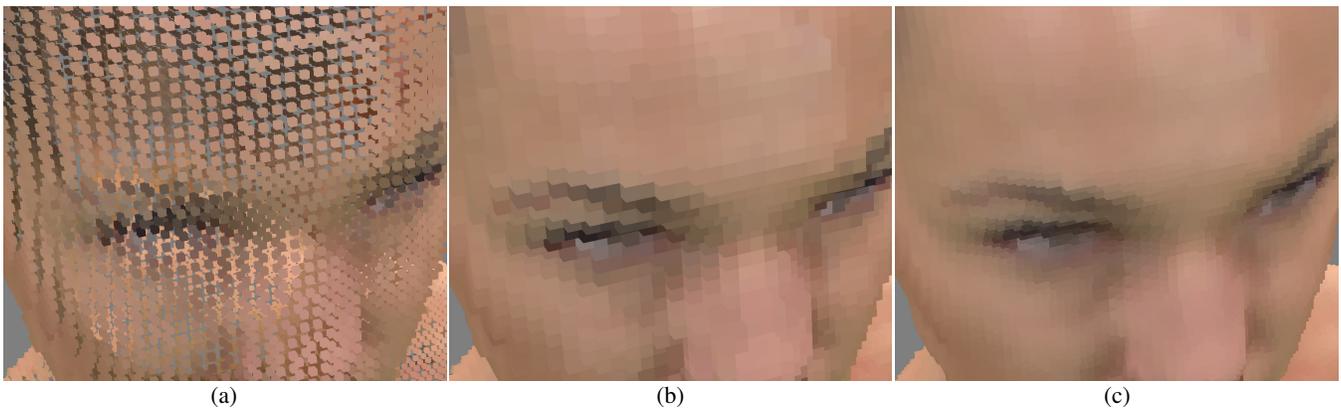


Fig. 6. *Longdress* compressed with $s = 2$: (a) compressed with G-PCC and expanded ($s \cdot V_d$); (b) compressed with G-PCC rendered with bigger voxels and (c) compressed with G-PCC+SR (V_{sr}).



Fig. 7. *Longdress* compressed with $s = 2$: (a) uncompressed (V^{orig}); (b) compressed with G-PCC (V_d) and (c) compressed with G-PCC+SR (V_{sr}).

TABLE II
BD-RATE COMPARISON TO G-PCC FOR TESTED PCs, IN RATE %. WE COMPARE THE PROPOSED METHOD TO (A) PCC-GEO-CNN-v2, (B) PCGCv2 AND (C) ADL-PCC.

Point cloud	A [23]	B [24]	C [25]	frac SR	frac SR (HR)
<i>dancer</i>	-77.04	-87.81	-93.63	-59.79	-91.77
<i>longdress</i>	-75.67	-77.06	-89.09	-62.39	-80.30
<i>loot</i>	-75.86	-74.88	-90.38	-64.40	-81.17
<i>queen</i>	-75.93	-79.46	-92.85	-70.23	-85.29
<i>redandblack</i>	-74.11	-75.88	-88.62	-62.52	-73.89
<i>soldier</i>	-76.99	-75.30	-89.39	-65.63	-80.66
<i>house</i>	-41.14	26.91	-87.77	-30.84	-
<i>statue_klimt</i>	-41.16	207.18	-39.50	-37.80	-

compress and to super-resolve. Part of this is due to the sparsity of the data, which makes it hard for the ML methods to converge, and decreases self-similarities at different scales. Also, those two non-solid PCs are notoriously noisy, which also contributes to the drop in performance.

The proposed method’s complexity arises primarily from neighborhood search, while ML methods typically demand GPU resources. Complexity comparisons among them present challenges as the order of magnitude of operations are quite different and are executed in different architectures, yet we expect the proposed method to be simpler and potentially

much faster execution when compared to ML approaches.

In Fig. 6, we compare the solutions to render the downsampled geometry from Eq. (1) using a viewpoint from *longdress* for $s = 2$. In Fig. 6(a), we see the expanded downsampled geometry ($s \cdot V_d$), without increasing its voxels sizes. In Fig. 6(b), we have the same geometry of Fig. 6(a), but voxels are now rendered with double their the original size to fill in the gaps. Finally in Fig. 6(c), we show V_{sr} , where one can easily see the refinement provided by the proposed solution. A zoomed out version comparing the original PC with V_d and V_{sr} is shown in Fig 7.

IV. CONCLUSIONS

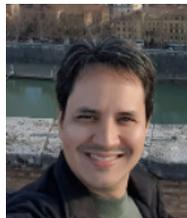
We have presented a method for post-processing PCs which were encoded and decoded using MPEG’s G-PCC codec by applying a previously proposed fractional SR technique. We compared this method with the originally decoded clouds as well as with ML-based end-to-end coding techniques. The results have shown that our method brings great improvement in quality (D1 PSNR metric) over G-PCC, specially for higher rate values (small scaling factor) in solid PCs. In all tested cases, the method is compatible to those based on ML, except for one, which outperforms all others. However, our method does not need any sort of previous training, which is required for the ML techniques.

REFERENCES

- [1] S. Schwarz and M. Pesonen, "Real-Time Decoding and AR Playback of the Emerging MPEG Video-Based Point Cloud Compression Standard," in *IBC*, 2019.
- [2] R. Mekuria, K. Blom, and P. Cesar, "Point Cloud Codec for Tele-immersive Video," ISO/IEC JTC1/SC29 WG7 (MPEG 3D Graphics Coding), Tech. Rep. m38136, 2016.
- [3] D. Flynn and S. Lasserre, "PCC Cat3 test sequences from BlackBerry—QNX," ISO/IEC JTC1/SC29 WG7 (MPEG 3D Graphics Coding), Tech. Rep. m43647, 2018.
- [4] C. Tulvan and M. Preda, "Point cloud compression for cultural objects," ISO/IEC JTC1/SC29 WG7 (MPEG 3D Graphics Coding), Tech. Rep. m37240, 2015.
- [5] 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, no. 1, 2020.
- [6] 3DG, "G-PCC codec description," ISO/IEC JTC1/SC29 WG7 (MPEG 3D Graphics Coding), Tech. Rep. N0151, 2021.
- [7] S. Lasserre and D. Flynn, "Improved TM3 for lossless and lossy compression of category 2 content," MPEG, Tech. Rep. m42519, 2018.
- [8] D. R. Freitas, E. Peixoto, R. L. de Queiroz, and E. Medeiros, "Lossy point cloud geometry compression via dyadic decomposition," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020. doi: 10.1109/ICIP40778.2020.9190910 pp. 2731–2735.
- [9] D. C. Garcia, T. A. Fonseca, R. U. Ferreira, and R. L. de Queiroz, "Geometry coding for dynamic voxelized point clouds using octrees and multiple contexts," *IEEE Trans. on Image Processing*, vol. 29, no. 1, pp. 313–322, Jan. 2020.
- [10] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, "Computing and rendering point set surfaces," *IEEE Trans. Vis. Comput. Graphics*, vol. 9, no. 1, pp. 3–15, 2003.
- [11] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. R. Zhang, "Edge-Aware Point Set Resampling," *ACM Trans. Graph.*, vol. 32, no. 1, Feb. 2013.
- [12] A. Hamdi-Cherif, J. Digne, and R. Chaine, "Super-Resolution of Point Set Surfaces Using Local Similarities," *Computer Graphics Forum*, vol. 37, no. 1, pp. 60–70, jun 2017.
- [13] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "PU-net: Point cloud upsampling network," in *IEEE/CVF Conf. on Comput. Vision and Pattern Recog. (CVPR)*. IEEE, jun 2018. doi: 10.1109/cvpr.2018.00295
- [14] C. Dinesh, G. Cheung, and I. V. Bajić, "3D Point Cloud Super-Resolution via Graph Total Variation on Surface Normals," in *IEEE Intl. Conf. Image Process. (ICIP)*. IEEE, sep 2019. doi: 10.1109/icip.2019.8803560
- [15] R. Li, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "PU-GAN: a point cloud upsampling adversarial network," in *IEEE Intl. Conf. on Computer Vision (ICCV)*, Oct. 2019. doi: 10.1109/ICCV.2019.00730
- [16] C. Dinesh, G. Cheung, and I. V. Bajić, "Super-Resolution of 3D Color Point Clouds Via Fast Graph Total Variation," in *IEEE Intl. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2020. doi: 10.1109/ICASSP40776.2020.9053971 pp. 1983–1987.
- [17] S. Ye, D. Chen, S. Han, Z. Wan, and J. Liao, "Meta-PU: An Arbitrary-Scale Upsampling Network for Point Cloud," *IEEE Trans. Vis. Comput. Graphics*, pp. 1–1, 2021.
- [18] T. M. Borges, D. C. Garcia, and R. L. de Queiroz, "Fractional Super-Resolution of Voxelized Point Clouds," *IEEE Trans. on Image Processing*, vol. 31, pp. 1380–1390, 2022.
- [19] D. Flynn and K. Mammou, "G-pcc ee13.46 review of v11 attribute coding," MPEG, Tech. Rep. m55485, 2020.
- [20] 3DG, "Common test conditions for G-PCC," ISO/IEC JTC1/SC29 WG7 (MPEG 3D Graphics Coding), Tech. Rep. N0094, 2021.
- [21] Y. Xu, Y. Lu, and Z. Wen, "Owlii Dynamic Human Textured Mesh Sequence Dataset," ISO/IEC MPEG JTC1/SC29/WG11, Macau, China, Tech. Rep. m41658, Oct. 2017.
- [22] E. d'Eon, B. Harrison, T. Myers, and P. A. Chou, "8i Voxelized Full Bodies, version 2 – A Voxelized Point Cloud Dataset," ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG), Geneva, input document m40059/M74006, January 2017.
- [23] M. Quach, G. Valenzise, and F. Dufaux, "Improved deep point cloud geometry compression," *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, 2020.
- [24] J. Wang, D. Ding, Z. Li, and Z. Ma, "Multiscale point cloud geometry compression," *2021 Data Compression Conference (DCC)*, pp. 73–82, 2021.
- [25] A. Guarda, N. M. M. Rodrigues, and F. Pereira, "Adaptive deep learning-based point cloud geometry coding," *IEEE Journal on Selected Topics in Signal Processing*, vol. 15, no. 2, pp. 415 – 430, Feb. 2021.
- [26] A. Zaghetto, "Performance Analysis of Currently AI-based Available Solutions for PCC," ISO/IEC JTC 1/SC 29/WG 7, Tech. Rep. N0174, 2021.
- [27] M. Quach, G. Valenzise, and F. Dufaux, "Learning convolutional transforms for lossy point cloud geometry compression," in *2019 IEEE international conference on image processing (ICIP)*. IEEE, 2019. doi: 10.1109/ICIP.2019.8803413 pp. 4320–4324.
- [28] D. Girardeau-Montaut, M. Roux, R. Marc, and G. Thibault, "Change detection on points cloud data acquired with a ground laser scanner," in *ISPRS Workshop Laser scanning*, G. Vosselman and C. Brenner, Eds., vol. XXXVI-3/W19. Enschede, NL: Intl. Society for Photogrammetry and Remote Sensing, Sep. 2005, pp. 30–35.



Tomás M. Borges, received the M.Sc. degree in electrical engineering in 2021 from the University of Brasilia (UnB). He is currently a Ph.D. student at UnB in Image and Video Processing focused on immersive content compression and quality evaluation for augmented reality systems.



Renan U. B. Ferreira, received the Electrical Engineer, M.Sc. and Ph.D. degrees in electrical engineering from the University of Brasilia, Brasilia, Brazil, in 2006, 2009 and 2015, respectively. He is currently a faculty member at the Gama College of Engineering of the University of Brasilia. His current research interests include image and video coding, biomedical image and point cloud processing.



Diogo C. Garcia, received the Electrical Engineer, M.Sc. and Ph.D. degrees in electrical engineering from the University of Brasilia, Brasilia, Brazil, in 2006, 2008 and 2012, respectively. He is currently a Faculty Member at the the Gama College of Engineering of the University of Brasilia. His current research interests include image and video coding, super resolution, and multiview and 3-D processing.



Ricardo L. de Queiroz, received the Ph.D. degree from The University of Texas at Arlington, in 1994. In 1990-1991, he was with the DSP research group at Universidade de Brasilia, as a research associate. He joined Xerox Corp. in 1994, where he was a member of the research staff until 2002. In 2000-2001 he was also an Adjunct Faculty at the Rochester Institute of Technology. He joined the Electrical Engineering Department at Universidade de Brasilia in 2003. In 2010, he became a Full (Titular) Professor at the Computer Science Department at Universidade de Brasilia. During 2015 he has been a Visiting Professor at the University of Washington, in Seattle. Dr. de Queiroz has published extensively in Journals and conferences and contributed chapters to books as well. He also holds 46 issued patents. His research interests include image and video compression, point cloud compression, multirate signal processing, and color imaging. Dr. de Queiroz is a Fellow of IEEE and a member of the Brazilian Telecommunications Society.