EXAMPLE-BASED SUPER-RESOLUTION FOR POINT-CLOUD VIDEO

Diogo C. Garcia, Tiago A. Fonseca and Ricardo L. de Queiroz

Universidade de Brasilia
Brasilia, Brasil
{*diogo,tiago*}*@image.unb.br and queiroz@ieee.org*

ABSTRACT

We propose a mixed-resolution point-cloud representation and an example-based super-resolution framework, from which several processing tools can be derived, such as compression, denoising and error concealment. By inferring the high-frequency content of low-resolution frames based on the similarities between adjacent full-resolution frames, the proposed framework achieves an average 1.18 dB gain over low-pass versions of the point-cloud, for a projection-based distortion metric [1, 2].

Index Terms— Point-cloud processing, 3D immersive video, free-viewpoint video, octree, super-resolution (SR).

1. INTRODUCTION

Recent demand for AR/VR applications have accelerated the interest in electronic systems to capture, process and render 3D signals such as point clouds [3, 4]. Nonetheless, there are no established standards regarding the capture, representation, compression and quality assessment of point clouds (PC). This lack of standards attracted attention to this research field and motivated a sequence of recent advances in processing 3D signals.

These signals can be captured using a set of RGBD cameras and represented as a voxelized point cloud. A voxelized PC consists in a set of points (x, y, z) constrained to lie on a regular 3D grid [5]. Each point can be considered as the address of a volumetric element, or voxel, which is said to be occupied or unoccupied and, for each occupied position, the surface color (RGB) is recorded. Instead of using a dense volumetric signal with all possible RGB samples for each frame, the point-cloud can be represented by a list of occupied voxels (geometry) and its color attributes, from now on referred as point-cloud.

A very efficient geometry representation can be obtained using the octree method, where the 3D space is recursively divided into fixed-size cubes, or octants, allowing for data compression, fast searching and spatial scalability [6, 7, 8]. Mixed-resolution scenarios (interleaved low- and full-resolution frames) naturally emerge from this representation: a low-resolution error-protected base layer, for instance, can be enhanced from a previously decoded full-resolution frame [9, 10]. The super-resolution technique was already explored when processing 3D signals, with works trying to increase data resolution in depth maps [11, 12, 13]. They usually explore geometric properties to improve the level of detail of a depth map. The contribution brought by this work is to infer high-frequency content (detail) of a point cloud by exploring the similarities between time-adjacent frames of already voxelized point-cloud signals as illustrated in Fig. 1. A concise signal model is derived in Section 2 and drives an example-based super-resolution framework, which is able to enhance the point-cloud level of detail as shown in Section 3. The conclusions are presented in Section 4.

Work partially supported by CNPq under grant 308150/2014-7. This paper is also available as arXiv:1803.06466 [eess.SP].

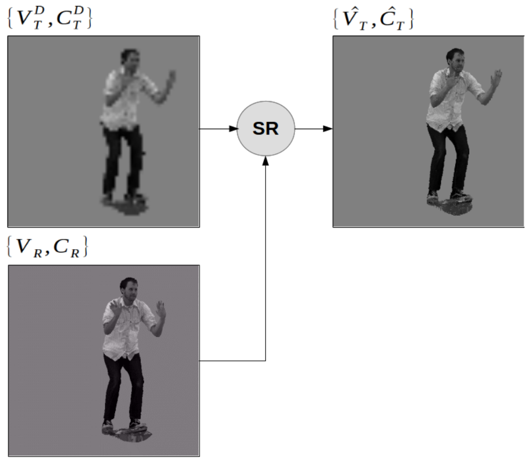


Fig. 1. The super-resolution framework (SR) outputs a super-resolved frame $\{\hat{\mathbf{V}}_T, \hat{\mathbf{C}}_T\}$ by exploring the high-frequency content similarities between a down-sampled point-cloud $\{\mathbf{V}_T^D, \mathbf{C}_T^D\}$ and a time-adjacent frame $\{\mathbf{V}_R, \mathbf{C}_R\}$.

2. PROPOSED METHOD

Traditional super-resolution (SR) [9] techniques generate high-resolution images from either multiple low-resolution images (multi-image SR) or databases of low- and high-resolution image pairs (example-based SR). The proposed method borrows ideas from the latter in order to increase the resolution and the details of a low-resolution point-cloud. Instead of relying on a database of low- and high-resolution point-cloud pairs, the proposed method extracts high-resolution information from an adjacent frame in a mixed-resolution point-cloud video sequence.

In this scenario, each point-cloud frame is composed of a pair of lists $\{\mathbf{V}, \mathbf{C}\}$, representing the XYZ positions of the occupied voxels and their corresponding color channels, respectively. Each down-sampled target frame $\{\mathbf{V}_T^D, \mathbf{C}_T^D\}$ is preceded by a full-resolution reference frame $\{\mathbf{V}_R, \mathbf{C}_R\}$. The proposed method generates the super-resolved version of the target frame, $\{\hat{\mathbf{V}}_T, \hat{\mathbf{C}}_T\}$, by adding the high-resolution information from the reference frame to the downsampled target frame.

In order to extract high-resolution information from $\{\mathbf{V}_R, \mathbf{C}_R\}$, the algorithm may estimate the motion of the voxels from one frame to another. However, only a down-sampled version of the target frame is available. By down-sampling the reference frame, $\{\mathbf{V}_R^D, \mathbf{C}_R^D\}$, motion can be estimated in a lower resolution of the point-clouds. However, the resolution of the motion estimation is also lowered. For example, if nearest-neighbour downsampling by a factor D_F of 2 is employed, the motion estimation is not be able to find any 1-voxel motion at full resolution, regardless of the motion direction.

A better estimation of the point-cloud's motion can be achieved by generating several downsampled versions of the reference frame considering incremental motion in all directions. For example, for $D_F = 2$, 8 downsampled versions can be generated by considering XYZ translations by $[0,0,0]$, $[0,0,1]$, $[0,1,0]$, $[0,1,1]$, $[1,0,0]$, $[1,0,1]$, $[1,1,0]$ and $[1,1,1]$. At full resolution, this is equivalent to dilating the reference point-cloud by a $2 \times 2 \times 2$ cube, rendering $\mathbf{V}_R^{DL}$. Figure 2 illustrates this concept in a 2D scenario, where only XY translations are considered, and dilation is performed with a 2×2 square.

Super-resolution of the target frame requires that $D_F \times D_F \times D_F$ voxels should be estimated for each voxel in $\mathbf{V}_T^D$. In order to do that, motion estimation is performed between $\mathbf{V}_T^D$ and $\mathbf{V}_R^{DL}$ on a voxel basis, using the $N \times N \times N$ neighborhood around each voxel as support. Each neighborhood $\mathbf{n}_T^D$ and $\mathbf{n}_R^{DL}$ is defined as a binary string, indicating which voxels are occupied and which are not. Since $\mathbf{V}_T^D$ and $\mathbf{V}_R^{DL}$ have different resolutions, the neighborhoods in these frames are obtained by skipping $D_F - 1$ positions around each of their voxels, which acts as the motion estimation between $\mathbf{V}_T^D$ and each of the downsampled versions of $\mathbf{V}_R$. Figure 3 illustrates this concept in a 2D scenario for a 3×3 neighborhood.

In the motion-estimation process, an $L \times L \times L$ search window around each voxel is chosen by the user, creating a tradeoff between speed and super-resolution accuracy. The cost function $C(i,j)$ between the i-th voxel in $\mathbf{V}_T^D$ and the j-th voxel in $\mathbf{V}_R^{DL}$ is defined as:

$$C(i,j) = \frac{H(i,j) + wD(i,j)}{w+1}, \tag{1}$$

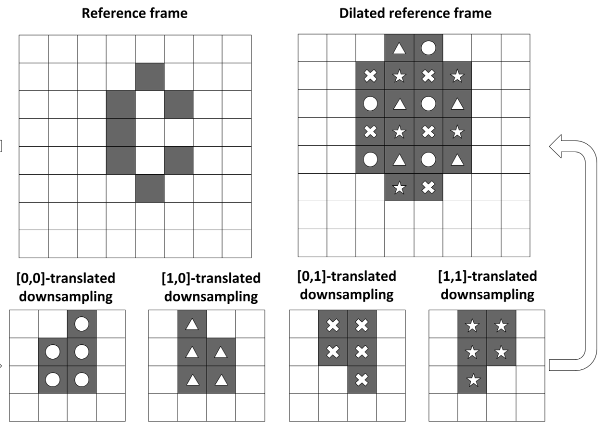

Reference frame
Dilated reference frame
[0,0]-translated downsampling
[1,0]-translated downsampling
[0,1]-translated downsampling
[1,1]-translated downsampling

Fig. 2. Downsampling of a 2D symbol by a factor of 2, considering 1-pixel translation in the XY directions, and posterior grouping, which can be seen as a dilation of the original symbol with a 2×2 square. The circles, squares, multiplication signs and stars illustrate from which downsampled version each position in the dilated version came from.

where $H(i,j)$ is the hamming distance between the neighborhoods $\mathbf{n}_T^D(i)$ and $\mathbf{n}_R^{DL}(j)$, $D(i,j)$ is the Euclidean distance between $\mathbf{V}_T^D(i)$ and $\mathbf{V}_R^{DL}(j)$, and w is the inverse of the Euclidean distance between the centers of mass of $\mathbf{V}_T^D$ and $\mathbf{V}_R^{DL}$. A small Euclidean distance between these centers of mass indicates small motion between the reference and target frames, increasing the value of w, i.e. $C(i,j)$ favors smaller motion vectors.

2.1. Performance Metrics

This work uses two measures to evaluate the achieved signal enhancements: a geometric quality metric [14], referred as GPSNR, and a distortion metric referred as projection peak signal to noise ratio (PPSNR) [1, 2]. GPSNR uses point-to-plane distances between point-clouds to calculate the geometric fidelity between them. The following steps are performed to evaluate the PPSNR:

1. Each target frame is expected to be represented in three versions: original $\{\mathbf{V}_T, \mathbf{C}_T\}$, low-pass $\{\mathbf{V}_T^{DL}, \mathbf{C}_T^{DL}\}$, and super-resolved $\{\hat{\mathbf{V}}_T, \hat{\mathbf{C}}_T\}$ point-clouds. For each version, project ($\pi_i(\{\mathbf{V}, \mathbf{C}\}) = f(x,y)$) the 3D signal on the faces of its surrounding cube to get 6 2D signals, $\Pi(\{\mathbf{V}, \mathbf{C}\}) = \bigcup_{i=1}^{6} \pi_i(\{\mathbf{V}, \mathbf{C}\})$, for each point-cloud version. Each projection $\pi_i(\{\mathbf{V}, \mathbf{C}\})$ outputs a 512×512 YUV [15] color image. Those represent the scene views from an orthographic projection on each cube face.

2. For each cube face, evaluate the luma PSNR between the projections of the original $\pi_i(\{\mathbf{V}_T, \mathbf{C}_T\})$ and the super-resolved $\pi_i(\{\hat{\mathbf{V}}_T, \hat{\mathbf{C}}_T\})$ signals and the projections of the original and the upsampled

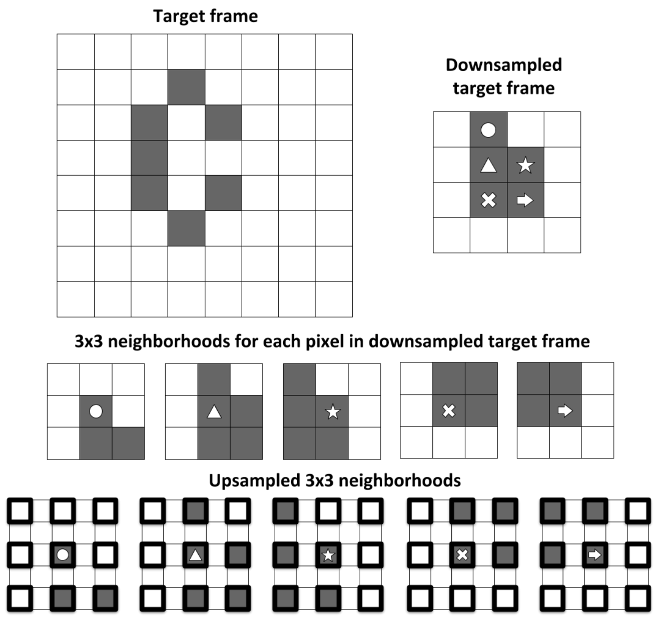

Target frame
Downsampled target frame
3x3 neighborhoods for each pixel in downsampled target frame
Upsampled 3x3 neighborhoods

Fig. 3. 3×3 neighborhood around the downsampled version of a 2D symbol by a factor of 2, and the upsampled version of these neighborhoods. Note that only every other position is considered in the upsampled version, as indicated by the thicker lines around the considered pixels.

$\pi_i(\{\mathbf{V}_T^{DL}, \mathbf{C}_T^{DL}\})$ signals. The PSNR between projections will provide 6 PSNRs values for each signal pair.

3. Average the 6 PSNRs to get two quality assessments: one evaluating the super-resolved and another evaluating the low-pass version. Then, subtracts the first value and the second to get the SR quality enhancement.

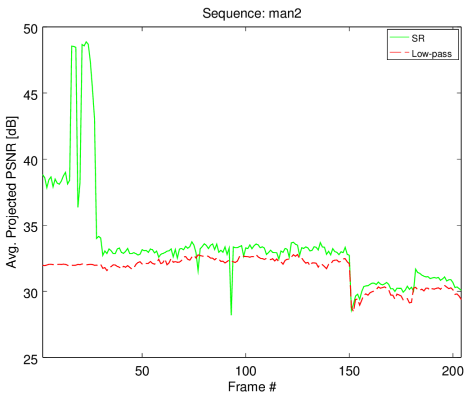
4. To get a PNSR value which represents the SR derived improvements, take the average of the PSNR differences along the sequence frames.

3. EXPERIMENTAL RESULTS

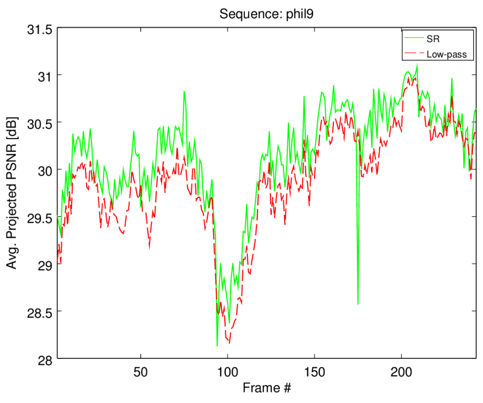
Tests for the proposed method were carried out with seven point-cloud sequences: *Andrew*, *David*, *Loot*, *Man*, *Phil*, *Ricardo* and *Sarah* [5, 16, 4]. The test set is made of five upper-body scenes of subjects captured at 30 fps and recorded at a spatial resolution of 512×512×512 voxels or a 9-level resolution. *Man* and *Loot* are full-body scenes recorded at 9- and 10-level resolution, respectively.The average PPSNR was calculated according to Sec. 2.1.

Table 1 summarizes the PPSNR performance of the proposed SR method for the test set. The comparison metric is the difference between the average PPSNR of the SR method and that evaluated for the low-pass version for each sequence. Table 2 shows the average PPSNR and GPSNR performance gains. [1] For all sequences, our method achieves a superior performance in inferring the high-frequency. The *Man* sequence benefited the most from the inferred high-frequency, while *Phil* achieved the most modest enhancement. The observed trend is that the more complex[2] the geometry, the greater potential to infer the high-frequency.

Figures 4(a) and (b) present the PPSNR on a frame-by-frame basis for the low-pass point-cloud version and for the SR version for sequences *Man* and *Phil*, respectively. Figure 4(a) shows the best high-frequency inference observed in the test set. This is due to a relative lack of motion in sequence *Man* on its first 30 frames; on the other side, an abrupt scene change around the 150th frame penalizes the quality of both versions (low-pass and SR). Despite the more challenging scenario in Fig. 4(b), the SR framework yields better enhancement than the low-pass signal, on average.



(a)



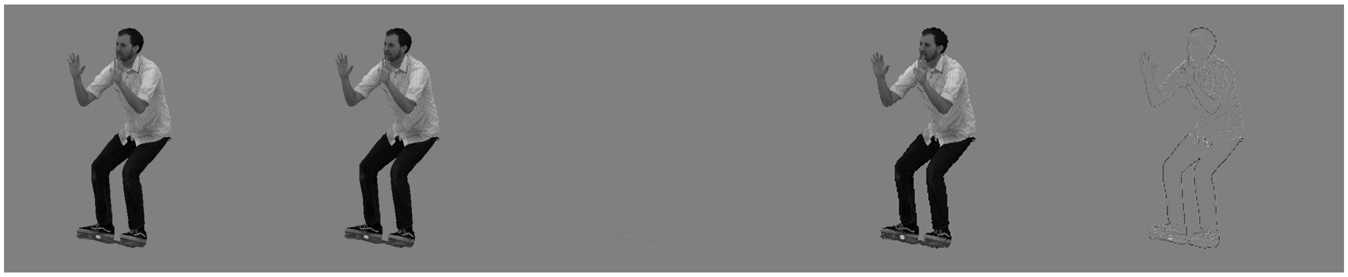
(b)

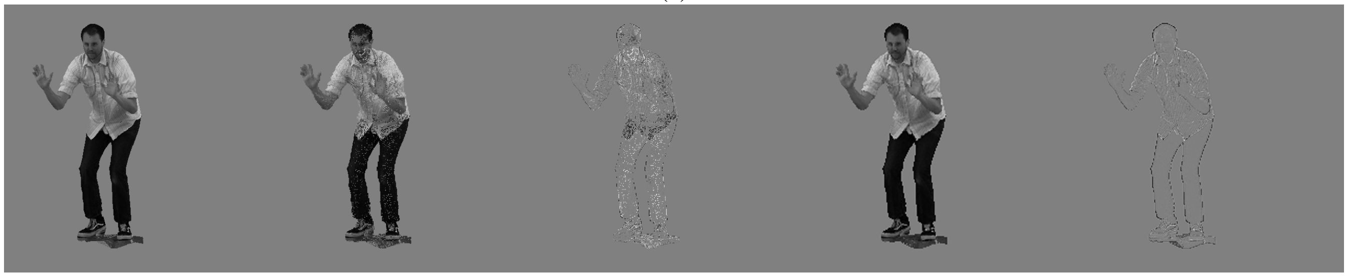
Fig. 4. Average PPSNR on a frame basis for SR and for the upsampled version (low-pass). Sequences: (a) *Man*; (b) *Phil*.

Figures 5(a)-(c) allow for the subjective evaluation of some point-cloud projections for sequences *Man* and *Phil*. Figure 5(a) shows the best SR performance for the test set,

[1]The average gains for *Loot* considers only its first 50 frames. For some frames of *Man*, the SR inserted geometric artifacts orthogonal to the PC normals, perfectly recovering the geometry in a point-to-plane sense [14]. This resulted in infinite GPSNR.

[2]In terms of bits per voxel to encode the full octree [17].


(a)


(b)

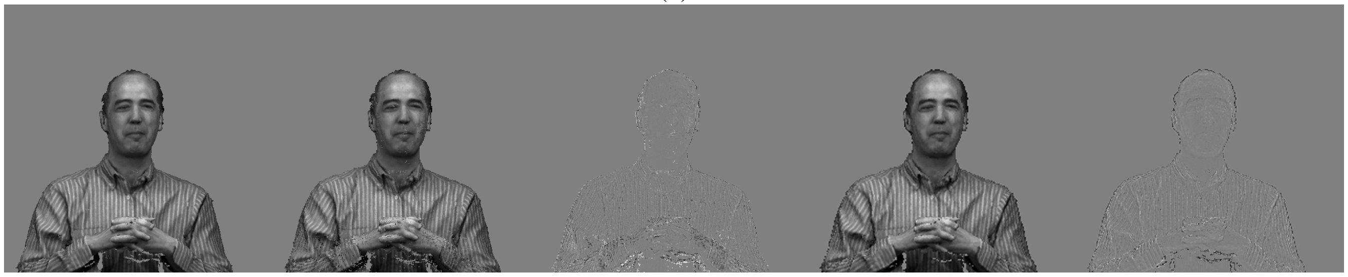

(c)

Fig. 5. Point-cloud projections for sequences (a)-(b) *Man*, frames 23 and 93, and (c) *Phil*, frame 175. For each image, from left to right, the columns correspond to the projections of: the original signal, the super-resolved signal, the residue of the super-resolved signal, the low-pass signal and the residue of the low-pass signal.

with an average 16.9 dB PPSNR gain over the low-pass version, mainly due to low movement of the test subject. The worst performance can be seen in Figs. 5(b) and (c), with average PPSNR losses of 4.12 and 1.84 dB over their low-pass versions, respectively.

Table 1. SR performance results. PPSNR-SR and PPSNR-LP stand for the average projected PSNR of the SR signal and the low-pass version, respectively. All values are in dB.

Sequence	PPSNR-SR	PPSNR-LP
Andrew	30.83	30.17
David	30.91	29.90
Loot	41.61	39.73
Man	33.52	31.59
Phil	30.15	29.89
Ricardo	33.60	32.36
Sarah	31.90	30.72

4. CONCLUSIONS

In this paper, an example based super-resolution framework to infer the high-frequency content of a voxelized point-cloud was presented. Based on an already efficient point-cloud representation [6], we benefited from its inherent scalability in resolution to explore similarities between point-cloud frames of test sequences. Experiments carried with seven point-cloud sequences show that the proposed method is able to successfully infer the high-frequency content for all the test sequences, yielding an average improvement of 1.18 dB when compared to a low-pass version of the test sequences. These results can benefit a point-cloud encoding framework, for efficient transmission, error concealment or even storage.

Table 2. SR performance improvements. PPSNR Gains and GPSNR Gains stand for the average gains in projected PSNR and in geometric quality metric [14, 2], respectively. All values are in dB.

Sequence	**PPSNR Gains**	**GPSNR Gains**
Andrew	0.76	4.99
David	1.01	4.25
Loot	1.84	5.40
Man	1.93	∞
Phil	0.27	4.61
Ricardo	1.24	5.16
Sarah	1.18	4.64
Average	**1.18**	**4.84**

5. REFERENCES

[1] R. L. De Queiroz, E. Torlig, and T. A. Fonseca, "Objective metrics and subjective tests for quality evaluation of point clouds," *ISO/IEC JTC1/SC29/WG1 input document M78030*, January 2018.

[2] R. L. de Queiroz and P. A. Chou, "Motion-compensated compression of dynamic voxelized point clouds," *IEEE Trans. on Image Processing*, vol. 26, no. 8, pp. 3886–3895, August 2017.

[3] S. Orts-Escolano et al., "Holoportation: Virtual 3d teleportation in real-time," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 2016, UIST '16, pp. 741–754.

[4] A. Collet et al., "High-quality streamable free-viewpoint video," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 69:1–69:13, Jul 2015.

[5] C. Loop, Q. Cai, S.O. Escolano, and P.A. Chou, "Microsoft voxelized upper bodies - a voxelized point cloud dataset," *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document m38673/M72012*, May 2016.

[6] D. Meagher, "Geometric modeling using octree encoding," *Computer Graphics and Image Processing*, vol. 19, no. 2, pp. 129–147, Jun 1982.

[7] R. L. de Queiroz and P. A. Chou, "Compression of 3d point clouds using a region-adaptive hierarchical transform," *IEEE Trans. on Image Processing*, vol. 25, no. 8, pp. 3497–3956, August 2016.

[8] R. L. de Queiroz, D. C. Garcia, P. A. Chou, and D. A. Florencio, "Distance-based probability model for octree coding," *IEEE Signal Processing Letters*, vol. to appear, 2018.

[9] E. M. Hung, R. L. de Queiroz, F. Brandi, K. F. Oliveira, and D. Mukherjee, "Video super-resolution using codebooks derived from key frames," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 22, no. 9, pp. 1321–1331, September 2012.

[10] E. M. Hung, D. C. Garcia, and R. L. de Queiroz, "Example-based enhancement of degraded video," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1140–1144, Sept 2014.

[11] D. B. Mesquita, M. F.M. Campos, and E. R. Nascimento, "A methodology for obtaining super-resolution images and depth maps from RGB-D data," in *Proc. Conference on Graphics, Patterns and Images*, August 2015.

[12] S. A. Ganihar, S. Joshi, S. Setty, and U. Mudenagudi, "3d object super resolution using metric tensor and christoffel symbols," in *Proc 2014 Indian Conference on Computer Vision Graphics and Image Processing*, December 2014, pp. 87:1–87:8.

[13] Y. Diskin and V. K. Asari, "Dense point-cloud creation using superresolution for a monocular 3d reconstruction system," in *Proc. SPIE 8399*, May 2012, vol. 8399, pp. 83990N1–83990N9.

[14] D. Tian, H. Ochimizu, C. Feng, R. Cohen, and A. Vetro, "Geometric distortion metrics for point cloud compression," in *Proc. IEEE Intl. Conf. Image Processing*, September 2017.

[15] I. E. Richardson, *H. 264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*, John Wiley & Sons, 2004.

[16] 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) input document m40059/M74006*, January 2016.

[17] D. C. Garcia and R. L. de Queiroz, "Context-based octree coding for point-cloud video," in *Proc. IEEE Intl. Conf. Image Processing*, September 2017.