#2273

Refining the Bounding Volumes for Lossless Compression of Voxelized Point Clouds Geometry

Emre Can Kaya* Sebastian Schwarz† Ioan Tabus*

*Tampere University, Tampere, Finland

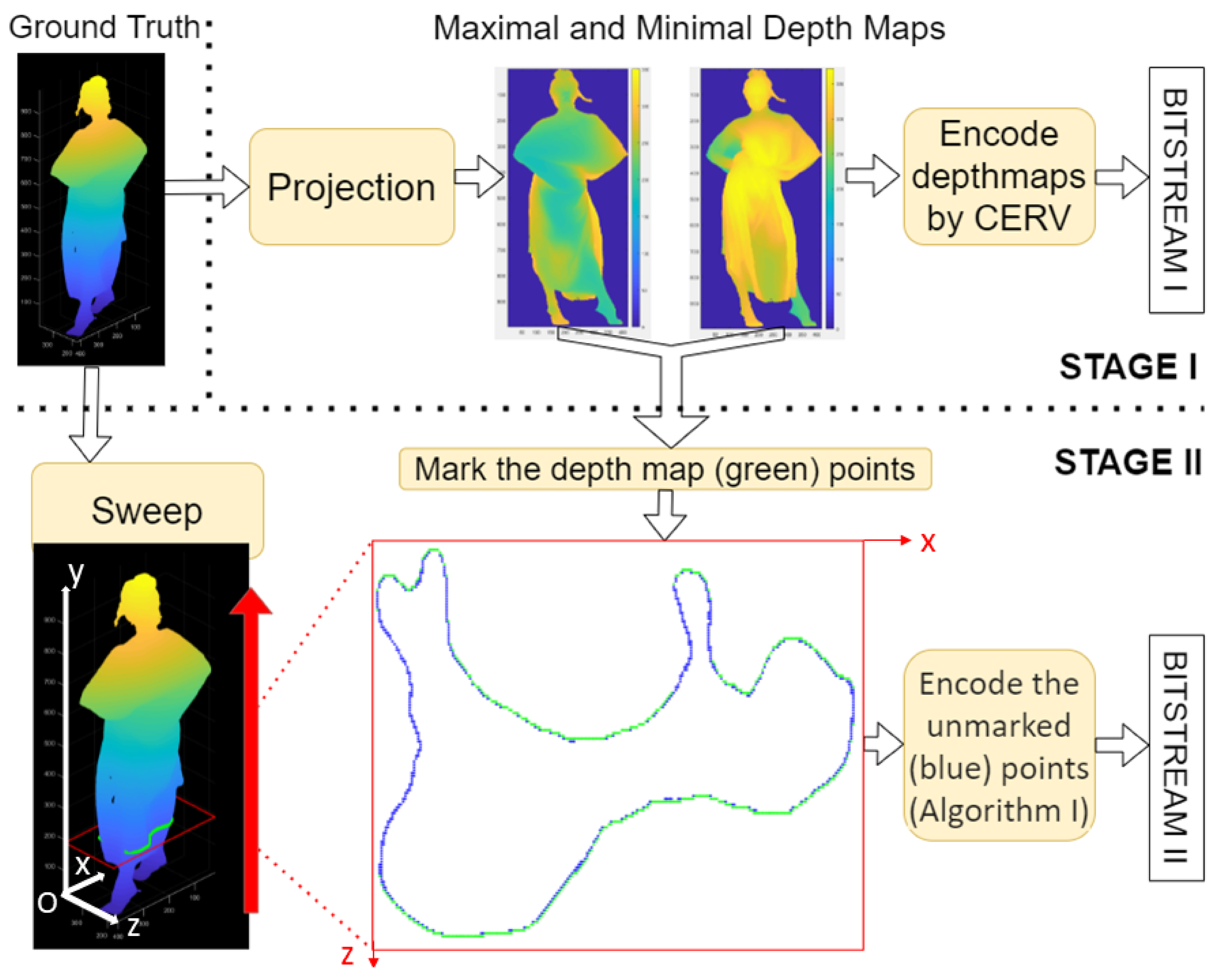
† Nokia Technologies, Munich, Germany

OVERVIEW

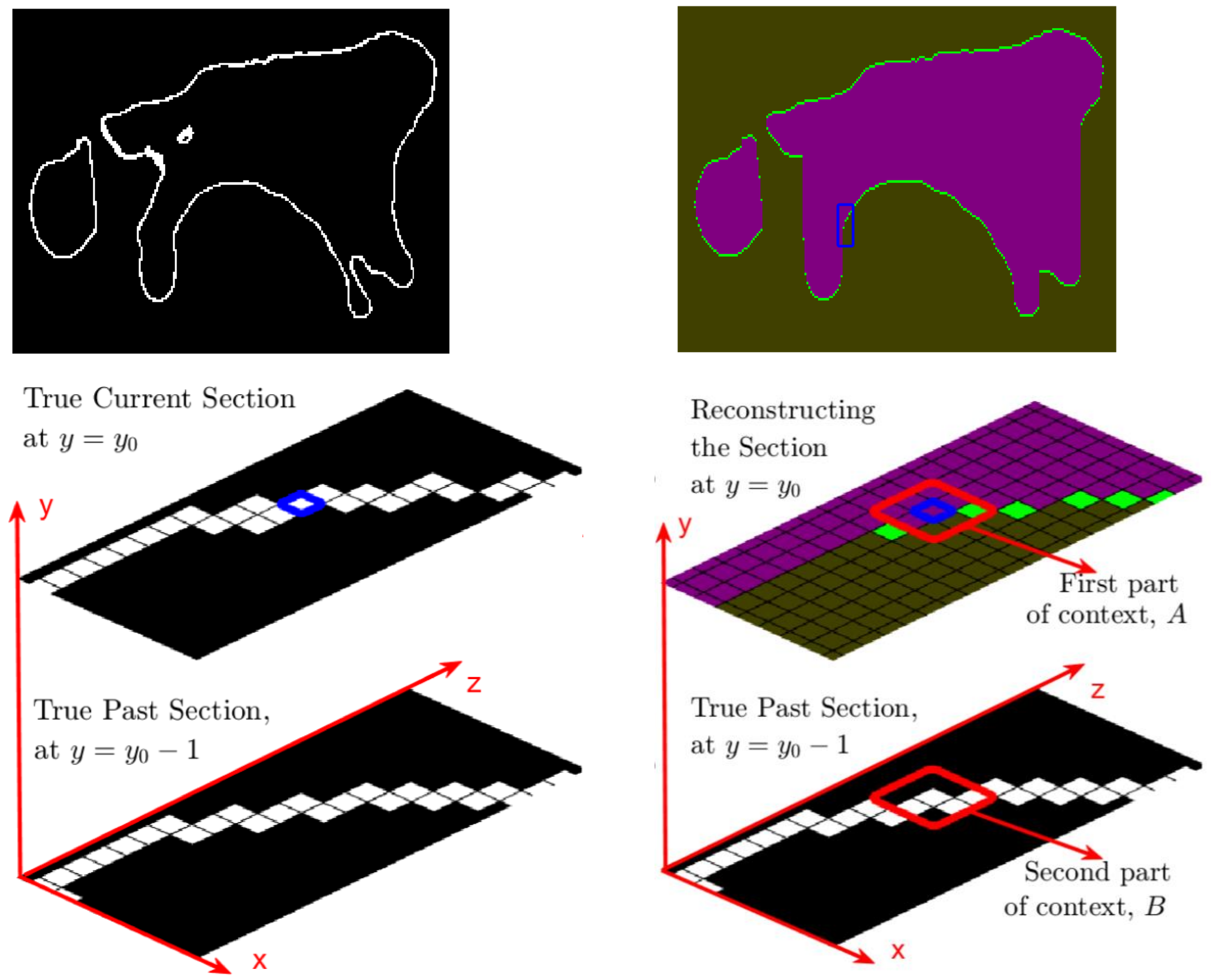
- A novel lossless compression method for point cloud geometry, related to [13] that aimed at reconstructing only the bounding volume of a point cloud. It consists of two stages:
- **Stage I:** Partially reconstruct the geometry from the two depthmaps associated to a single projection direction.
- **Stage II:** Complete it to a full reconstruction by sweeping section by section along one direction and encoding the points which were not contained in the two depthmaps.
- One main ingredient is a list-based encoding of the inner points (situated inside the feasible regions) by a novel arithmetic 3D context coding procedure that efficiently utilizes rotational invariances present in the input data.
- State-of-the-art bits-per-voxel results on benchmark datasets

PROPOSED METHOD

1) Block diagram of the overall scheme



2) Illustrating the method for encoding the current voxel

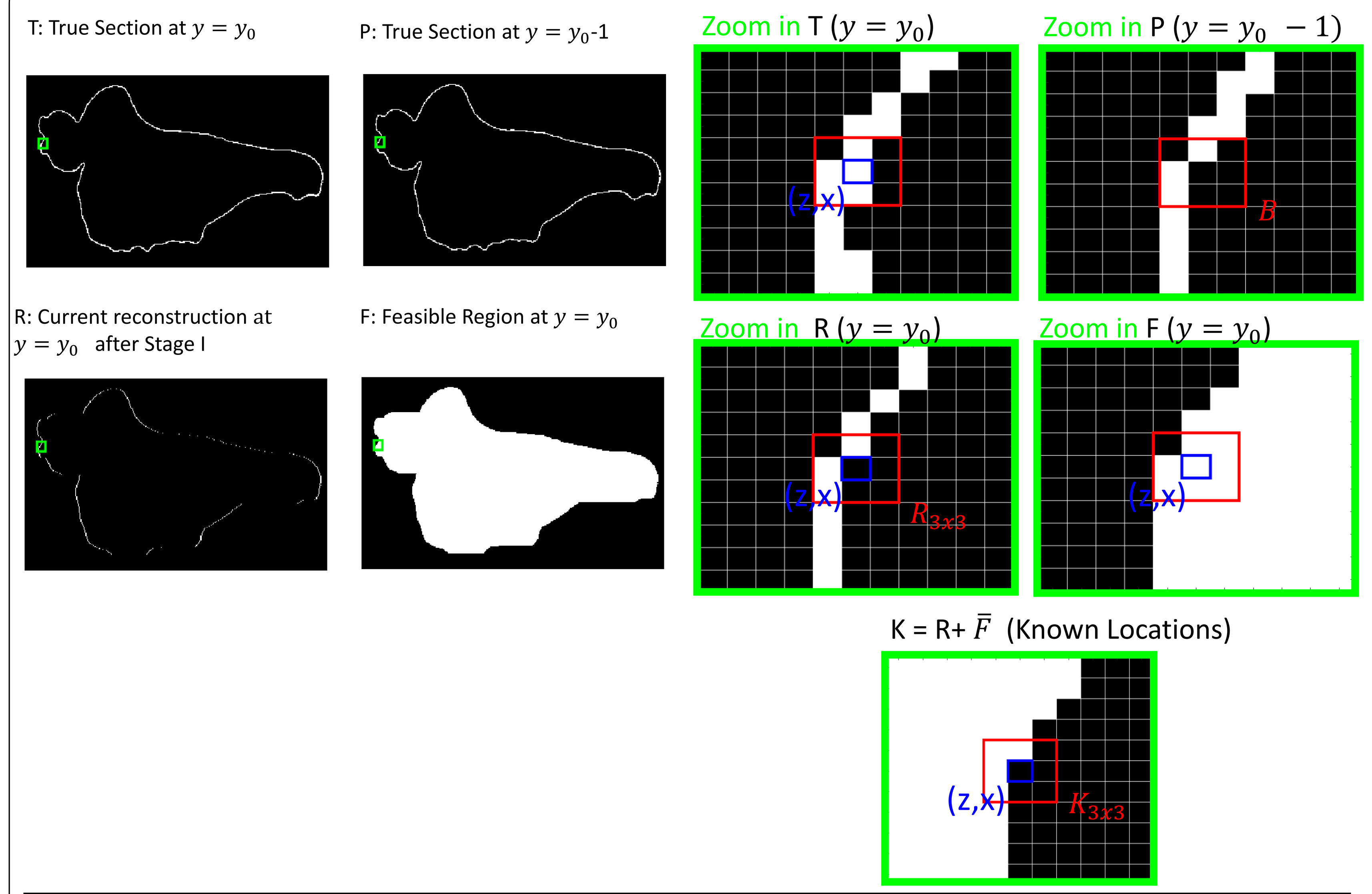


Algorithm 1 Stage II of encoding

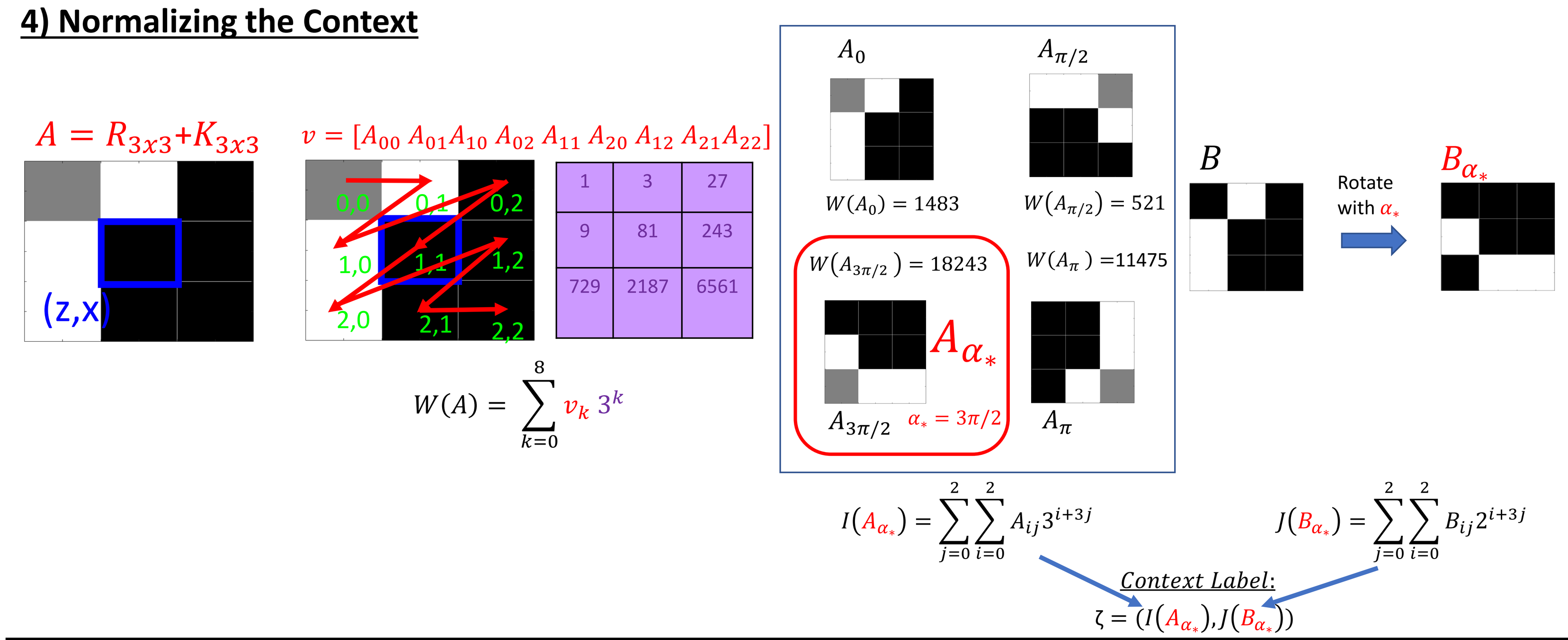
Require: T: True section binary image at $y = y_0$ (N_z x N_x)
P: True section binary image at $y = y_0 - 1$ (N_z x N_x)
R: Current reconstruction at $y = y_0$ (N_z x N_x)
F: Feasible Regions bin. image derived from R (Sect. 2.2)
K: Binary image of known locations $K \leftarrow \overline{F} + R$
L: Pixels to be processed $L \leftarrow \{(z, x) \ni R(z, x) = 1\}$
M: Binary image of pixels that has been to L. $M \leftarrow$R

while $L \neq \emptyset$ **do**
Read (z, x) from the top of L
Extract a 3x3 matrix R_{3x3} from R centered at (z, x)
Extract a 3x3 matrix B from P centered at (z, x)
Extract a 3x3 matrix K_{3x3} by cropping K around (z, x)
$A \leftarrow R_{3x3} + K_{3x3}$
Find normalizing rotation $\alpha^*(A)$ and form A_{α^*}
Use $\alpha^*(A)$ to rotate B as B_{α^*}
Form the context $\zeta = (I(A_{\alpha^*}), J(B_{\alpha^*}))$
Encode $T(z, x)$ using the context ζ
if $T(z, x) == 1$ **then**
Append to L every neighbor (z_n, x_n) of (z, x) for which $K(z_n, x_n) = 0$ and $M(z_n, x_n) = 0$
If (z_n, x_n) is appended, $M(z_n, x_n) \leftarrow 1$
end if
Update R: $R(z, x) \leftarrow T(z, x)$
Update K: $K(z, x) \leftarrow 1$
Remove (z, x) from L
end while

3) Defining the Binary Images for Extracting the Context at Pixel (z,x)



4) Normalizing the Context



5) Experimental Results

Table 1. Average Rate for the first 200 frames from MVUB [15] and 8i [16] datasets for the proposed Bounding Volume Lossless (BVL) encoder compared to recent codecs.

Sequence	Average Rate [bpv]			
	P(PNI)[8]	**TMC13[9]**	**DD[12]**	**BVL**
Microsoft Voxelized Upper Bodies [15]				
Andrew9	1.83	1.14	**1.12**	1.17
David9	1.77	1.08	**1.06**	1.10
Phil9	1.88	1.18	**1.14**	1.20
Ricardo9	1.79	1.08	**1.03**	1.05
Sarah9	1.79	**1.07**	**1.07**	1.08
Average	1.81	1.11	**1.08**	1.12
8i Voxelized Full Bodies [16]				
Longdress	1.75	1.03	0.95	**0.91**
Loot	1.69	0.97	0.91	**0.88**
Redandblack	1.84	1.11	**1.03**	**1.03**
Soldier	1.76	1.04	**0.96**	**0.96**
Average	1.76	1.04	0.96	**0.94**

TABLE 2
AVERAGE RATE [BPOV] RESULTS ON POINT CLOUDS FROM CAT1A

Point Cloud	Bitdepth	G-PCC	BVL
basket. player	11	0.885	0.852
dancer	11	0.876	0.826
facade 00064	11	1.1969	1.3331
queen	10	0.7817	0.7883
redandblack	10	1.1055	1.0418
loot	10	0.9818	0.8991

REFERENCES

[8] 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 Transactions on Image Processing, vol. 29, pp. 313–322, 2019.
[9] "MPEG Group TMC13,"https://github.com/MPEGGroup/mpeg-pcc-tmc13, Accessed: 2020-03-20.
[12] E. Peixoto, "Intra-frame compression of point cloud geometry using dyadic decomposition," IEEE Signal Processing Letters, vol. 27, pp. 246–250, 2020.
[13] I. Tabus, E. C. Kaya, and S. Schwarz, "Successive refinement of bounding volumes for point cloud coding," in 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP). IEEE, 2020, pp. 1–6.