Efficient Projected Frame Padding for Video-based Point Cloud Compression

Li Li, *Member, IEEE*, Zhu Li, *Senior Member, IEEE*, Shan Liu, *Senior Member, IEEE*, Houqiang Li, *Senior Member, IEEE*

***Abstract*—The state-of-the-art 2D-based dynamic point cloud (DPC) compression algorithm is the video-based point cloud compression (V-PCC) developed by the Moving Pictures Experts Group (MPEG). It first projects the DPC patch by patch from 3D to 2D and organizes the projected patches into a video. The video is then efficiently compressed by High Efficiency Video Coding. However, there are many unoccupied pixels that may have a significant influence on the coding efficiency. These unoccupied pixels are currently padded using either the average of 4-neighbors for the geometry or the push-pull algorithm for the color attribute. While these algorithms are simple, the unoccupied pixels are not handled in the most efficient way. In this paper, we divide the unoccupied pixels into two groups: those that should be occupied and those that should not be occupied according to the occupancy map. We then design padding algorithms tailored to each group to improve the rate-distortion performance of the V-PCC reference software, for both the geometry and the color attribute. The first group is the unoccupied pixels that should be occupied according to the block-based occupancy map. We attempt to pad those pixels using the real points in the original DPC to improve the quality of the reconstructed DPC. Additionally, we attempt to maintain the smoothness of each block so as not to negatively influence the video compression efficiency. The second group is the unoccupied pixels that were correctly identified as unoccupied according to the block-based occupancy map. These pixels are useless for improving the reconstructed quality of the DPC. Therefore, we attempt to minimize the bit cost of these pixels without considering their reconstruction qualities. The bit cost is determined by the residue of these pixels obtained by subtracting the prediction pixels from the original pixels. Therefore, we propose padding the residue using the average residue of the occupied pixels in order to minimize the bit cost. The proposed algorithms are implemented in the V-PCC and the corresponding HEVC reference software. The experimental results show the proposed algorithms can bring significant bitrate savings compared with the V-PCC.**

***Index Terms*—Frame padding, High Efficiency Video Coding, Occupancy map, Point cloud compression, Video-based point cloud compression**

Manuscript received July 11, 2019; revised March 07, 2020 and July 15, 2020; accepted August 9, 2020. This work was recommended by Associate Editor Dr. Cheung. Sen-Ching Samson. This work was supported in part by the National Key Research and Development Plan under Grant 2017YFB1002401. This work was supported in part by the NSF I/UCRC on Big Learning and Tencent Media Lab.

L. Li and H. Li are with the CAS Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, University of Science and Technology of China, Hefei 230027, China. Dr. Li Li is the corresponding author. (e-mail: lil1@ieee.org; lihq@ustc.edu.cn)

Z. Li is with the Department of Computer Science and Electrical Engineering, University of Missouri-Kansas City, MO 64110, USA. (e-mail: lizhu@umkc.edu).

S. Liu is with the Tencent America, 661 Bryant St, Palo Alto, CA 94301, USA. (e-mail: shanl@tencent.com).

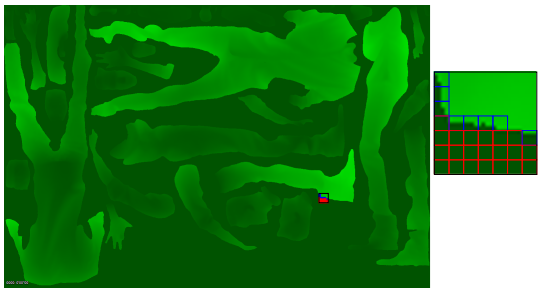
I. Introduction

A point cloud is a set of 3D points that can be used to represent a 3D surface. Each point has a spatial position (x, y, z) and a vector of attributes such as colors, normals, or material reflectance. In this paper, we assume the attributes are colors represented as (R, G, B) tuples in RGB color space. The point cloud is a convenient representation for reconstructing 3D objects or scenes in applications such as virtual reality [1] and autonomous navigation [2]. The point cloud is superior to the 360-degree video because it can support 6 degrees of freedom (DoF) rather than 3 DoF, which results in an enhanced user experience [3]. For a more thorough review of point cloud applications, refer to [4].

A popular emerging use of 3D point clouds is in representing human 'holograms' that can be used in virtual-reality, augmented-reality, and mixed-reality applications. Most recently, 8i captured several dynamic point clouds (DPCs) for this purpose using camera plus depth sensors [5]. Each frame of these DPCs has around one million points. Larger numbers of points generally support a higher-resolution representation of 3D objects, however, this also leads to serious transmission and storage burdens. With each point represented by 30 and 24 bits for the geometry and attribute, respectively, a single frame with one million points can be as large as $6M$ bytes without compression. For a DPC with 30 frames per second, the bitstream size without compression can be as high as $180M$ bytes per second. Therefore, there is an urgent need to compress the DPC more efficiently.

The DPC compression method can be roughly divided into two groups: the 3D-based method and the 2D-based method. As its name implies, the 3D-based method [6] [7] compresses the DPC directly in the 3D domain. 3D motion estimation (ME) and motion compensation (MC) are the key technologies in the 3D-based method for exploiting the inter correlations among various frames. However, neighboring frames may have a different number of points, and the points in neighboring frames do not have a one-to-one correspondence for ME and MC. Therefore, 3D ME and MC cannot efficiently characterize the inter correlations efficiently thus the compression performance is unsatisfactory.

The 2D-based method [8] [9] attempts to utilize the mature video compression standard for 2D videos by projecting the DPC onto a 2D video. The state-of-the-art 2D-based DPC compression is the video-based point cloud compression (V-PCC) [10] developed by the Moving Pictures Experts Group (MPEG). The V-PCC projects the DPC to the six faces of its



(a) Geometry

(b) Attribute

Fig. 1. Typical examples of the projected geometry and attribute frames derived from the V-PCC reference software without padding [12]. These frames are the first frames (picture order count 0) of the projected geometry and attribute video for the DPC "Loot". The magnified block is a 28×28 block. The size of the small squares inside the block is 4×4, which is the occupancy-map precision. There are two groups of unoccupied pixels in the magnified block. One group is the unoccupied pixels signaled as unoccupied as indicated by the red small squares. The other group is the unoccupied pixels signaled as occupied as indicated by the blue small squares.

bounding box patch by patch and packs all the patches into a video to utilize the High Efficiency Video Coding (HEVC) [11]. The V-PCC achieves a good balance between the bitrate of the packed video and the quality of the reconstructed DPC. However, it leads to many unoccupied pixels among the patches. Those unoccupied pixels have a significant negative influence on the V-PCC compression efficiency.

Fig. 1 shows typical examples of the projected geometry and attribute frames derived from the V-PCC reference software without padding [12]. The size of the magnified block is 52×52. The size of the small squares inside the block is 4×4, which is the occupancy-map precision. The occupancy-map precision setting as 4×4 denotes that one bit is transmitted for each 4×4 block to indicate whether it is occupied. In the magnified block, we can see two types of unoccupied pixels depending on whether they are signaled as occupied or not according to the occupancy map. One type is the unoccupied pixels signaled as occupied, indicated by the small blue blocks in the magnified area. In a 4×4 block in the geometry frame, if there exist occupied pixels as indicated by light green, all the pixels including the unoccupied pixels as indicated by dark green are signaled as occupied. These unoccupied pixels affect the reconstructed DPC quality. They are called unoccupied-occupied pixels in the following sections. To guarantee that each reconstructed point has a valid attribute, the attribute video is recolored based on the reconstructed geometry. Therefore, there is not a concept of unoccupied-occupied pixels in the attribute video. The other type is the unoccupied pixels signaled as unoccupied, indicated by the small red blocks in the magnified area. These unoccupied pixels do not affect the reconstructed point cloud quality. They are called unoccupied-unoccupied pixels in the following sections. Without padding, both the unoccupied-occupied and unoccupied-unoccupied pixels significantly break the spatial correlations. Therefore, they significantly affect the performance of V-PCC if not properly padded.

For the unoccupied-occupied pixels, as they influence the reconstructed DPC quality, we need to simultaneously consider their bit cost and the reconstructed DPC quality when they are padded. The current padding methods on unoccupied-occupied pixels either consider the bit cost [12] or the reconstructed DPC quality [13]. Therefore, they cannot achieve a satisfactory rate-distortion (RD) performance. For the unoccupied-unoccupied pixels, as they have no influence on the reconstructed DPC quality, the optimization target is to minimize their bit cost. Various padding methods [14] [15] are proposed to keep the original block smooth to reduce their bit cost. However, in HEVC, the bit cost of a prediction unit (PU) is essentially determined by the smoothness of the residual block obtained through subtracting the prediction block from the original block. Therefore, without an explicitly-optimized algorithm to minimize the bit cost of the residual block, it is impossible to achieve a satisfactory RD performance.

To address the above problems, we propose a modified V-PCC framework with two advanced padding methods to deal with the unoccupied-occupied and unoccupied-unoccupied pixels, respectively. Our contributions in this paper are summarized as follows.

- We propose padding the unoccupied-occupied pixels with the real points from the original DPC considering the reconstructed DPC quality. In addition, we require the padded points to be similar to the occupied pixels in the same block to retain its smoothness. This scheme simultaneously considers the bit cost and the reconstructed DPC quality, and thus leads to significant performance improvements.
- To minimize the bit cost of the unoccupied-unoccupied pixels, we propose a padding algorithm to smooth the residual block instead of the original block. For a PU with both occupied and unoccupied-unoccupied pixels, we first obtain the residue of the occupied pixels by subtracting the prediction from the original. Then the average of the residue of all the occupied pixels is used to pad the residue of the unoccupied-unoccupied pixels. Therefore, the residue of the PU can have a small variation, and thus can be coded with a small number of bits.
- We implement the proposed algorithms in the V-PCC and the corresponding HEVC reference software. The proposed two algorithms can individually lead to significant coding gains. They can also be combined into a unified padding framework to achieve an even better RD performance.

The rest of this paper is organized as follows. Section II

introduces the related work on DPC compression and the state-of-the-art padding methods. The proposed algorithms including the unoccupied-occupied padding and unoccupied-unoccupied padding are described in Section III. In Section IV, detailed experimental results are shown. Section V concludes the paper.

II. Related work

In this section, we first introduce the 3D-based and 2D-based methods on DPC compression. Then we introduce the recent work on padding methods of unoccupied-occupied and unoccupied-unoccupied pixels under the V-PCC framework.

A. Dynamic point cloud compression

1) 3D-based method: Kammerl *et al.* [16] proposed encoding the geometry differences for exploiting the inter correlations from frame to frame. However, this method is limited since there is not always a one-to-one point correspondence between the neighboring frames. Thanou *et al.* [6] proposed that the correlations between neighboring frames could be better characterized in the spectral domain rather than in the pixel domain and performed 3D ME by feature matching. However, some points may be unable to find the corresponding points using this global method. To solve this problem, Queiroz and Chou [7] proposed dividing the point cloud into multiple cubes and performed ME and MC cube by cube. However, the 3D translational motion model used could not efficiently characterize the 3D motion. Mekuria *et al.* [17] proposed using an iterative closest point (ICP) method to employ high-order motion models to better characterize the motions. While a number of methods have been proposed to improve the efficiency of the 3D-based method, they only partially alleviate the problems of 3D ME and MC. Due to inflexible partitions and inaccurate motion vector predictors, efficiently performing 3D ME and MC is still an issue with DPCs.

2) 2D-based method: Lasserre *et al.* [18] and Budagavi *et al.* [19] proposed sorting the points directly into a video using an octree or point position in a lossless manner. However, the videos generated were unsuitable for the current video compression framework since the spatial and temporal correlations of the videos were limited. To solve this problem, Schwarz *et al.* [8] and He *et al.* [20] proposed projecting the DPC to a cube or cylinder and unfolding it into a 2D video. The generated videos were easy to encode, however, many points were lost due to occlusion. To solve this problem, Mammou *et al.* [9] proposed a patch-based method to project the DPC to the cube patch by patch and organized all the patches into a video. This method was successful and won the MPEG call for proposals for DPC compression [21] in 2017. Consequently, it was named V-PCC by the MPEG group. Nonetheless, the projected video introduced many unoccupied pixels among the patches. These unoccupied pixels should be carefully handled to avoid a significant negative influence on the V-PCC compression efficiency.

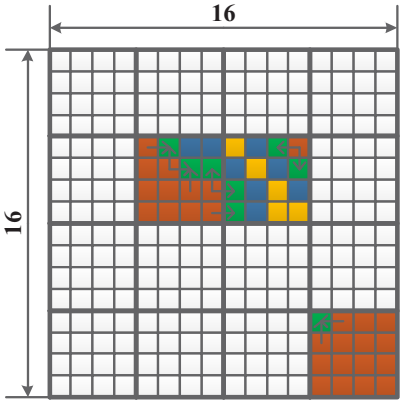


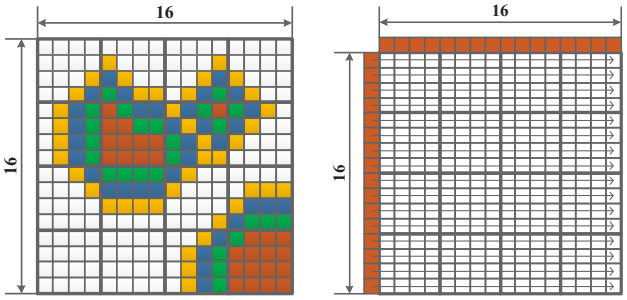
Fig. 2. Padding process of the unoccupied-occupied pixels in the V-PCC reference software [12]. The colored three 4×4 blocks are signaled as occupied. The orange small squares represent the occupied pixels.

B. Recent padding methods for V-PCC

1) Padding for the unoccupied-occupied pixels: Fig. 2 shows the padding process of the unoccupied-occupied pixels of a typical 16×16 geometry block in the V-PCC reference software [12]. Each small square represents a pixel. The three colored 4×4 blocks are signaled as occupied. The orange small squares are the occupied pixels. The unoccupied-occupied pixels are iteratively padded using the 4-neighbor occupied pixels. The green pixels are first padded using the average of the 4-neighbor orange pixels as indicated by the arrows. The blue pixels are then padded using the average of the 4-neighbor green pixels. Finally, the yellow pixels are padded using the blue pixels. This method creates a smooth block that can be efficiently compressed. However, the padded pixels generate some 3D points that do not correspond to any points in the original DPC. This leads to an increase in distortions, particularly on point-to-plane errors. In addition, Graziosi and Tabatabai [22] proposed using the uncompressed full-resolution occupancy map to guide the padding process. Ke *et al.* [13] introduced patch expansion in the boundary to avoid points loss. These methods considered only the quality of the reconstructed DPC while ignoring the smoothness of the padded block. As a result, they led to a serious bitrate increase compared with the current method in the V-PCC reference software.

2) Padding for the unoccupied-unoccupied pixels: The padding for unoccupied-unoccupied pixels in the V-PCC reference software [12] can be divided into two groups: geometry padding and attribute padding. The geometry padding is based on a 16×16 block as shown in Fig. 3. If part of the 16×16 block is occupied as shown in Fig. 3 (a), the 16×16 block is padded iteratively using the average of the 4-neighbor occupied pixels. If the whole 16×16 block is unoccupied as shown in Fig. 3 (b), the 16×16 block is padded from the neighboring pixels using horizontal copy or vertical copy if the horizontal neighboring pixels are not available.

As the attribute frame is less smooth compared with the geometry frame, more padding methods have been developed to improve the compression performance of the attribute video.



(a) Part unoccupied (b) Full unoccupied

Fig. 3. Padding process of the unoccupied-unoccupied pixels for the geometry in the V-PCC reference software [12]. The orange small squares in (a) represent the occupied pixels. The orange small squares in (b) represent the neighboring available pixels.

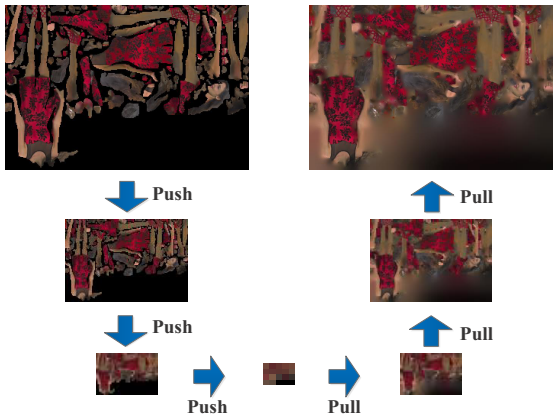


Fig. 4. Padding process of the unoccupied-unoccupied pixels using the push-pull algorithm for the attribute in the V-PCC reference software [12].

In the V-PCC reference software [12], the attribute padding is based on the push-pull algorithm as shown in Fig. 4 [14]. The push-pull algorithm [23] creates a multi-resolution representation of the attribute frame and fills in the unoccupied-unoccupied pixels with pixel values from lower levels. During the V-PCC standardization process, a sparse linear model-based method [24], a harmonic background filling method [25], and a smoothed push-pull algorithm [15] are further proposed to improve the padding of the attribute.

All these unoccupied-unoccupied padding methods for geometry and attribute are based on the original pixels and can guarantee the smoothness of each original block. However, they do not consider that the smoothness of the original block does not always lead to the smoothness of the residual block. In geometry and attribute videos, the corresponding blocks in neighboring frames may be put in totally different positions. Therefore, it is possible that the original block is smooth while the best-matching prediction block is unsmooth. Thus, the residual block obtained through subtracting the prediction block from the original block is unsmooth and leads to considerable bit cost. In this paper, we attempt to pad the unoccupied-unoccupied pixels to guarantee the smoothness of the residual block to achieve the least bit cost.

3) Padding for the other videos: There have been some works that focused on padding for general videos [26] or 360-degree videos [27]. Li *et al.* [26] proposed a padding method for the frame boundary for arbitrary resolution video coding. When the picture does not contain an integer number of macroblocks, some pixels are padded in a manner that minimizes the bitrate. The pixels are padded to force the residual of padded pixels to 0. However, this method only deals with padding regular structures near the picture boundary. Under the V-PCC framework, irregular structures within the picture need to be padded. Li *et al.* [27] proposed a padding method to maintain a continuous reference frame for the projected 360-degree video. This method is unsuitable for the V-PCC framework since it is designed for the reference frame without considering the bitrate of the padded pixels.

TABLE I
PERFORMANCE COMPARISON BETWEEN SETTING THE OCCUPANCY-MAP PRECISION AS 4×4 AND 1×1.

Test point clouds	Geom.BD-GeomRate D1	Geom.BD-GeomRate D2	Attr.BD-AttrRate Luma	Attr.BD-AttrRate Cb	Attr.BD-AttrRate Cr
Loot	95.1%	78.9%	–1.4%	–2.4%	–4.4%
RedAndBlack	98.5%	74.0%	–1.9%	–4.2%	–2.5%
Soldier	115.0%	97.7%	–4.9%	–11.6%	–8.8%
Queen	129.6%	103.7%	–9.5%	–12.6%	–9.8%
LongDress	90.0%	72.1%	–2.0%	–2.7%	–1.5%
Avg.	**105.6%**	**85.3%**	**–3.9%**	**–6.7%**	**–5.4%**

III. PROPOSED ALGORITHMS

In this section, we introduce the proposed padding methods. The padding of unoccupied-occupied pixels is introduced in subsection III-A. The padding of the unoccupied-unoccupied pixels is introduced in subsection III-B.

A. Padding of unoccupied-occupied pixels

As mentioned in Section I, setting the occupancy-map precision as 4×4 is the reason why the unoccupied-occupied pixels exist. If we set the occupancy-map precision as 1×1, the problem of unoccupied-occupied pixels will not exist. Therefore, before introducing the proposed padding method for unoccupied-occupied pixels in the V-PCC reference software, we first compare the performance of setting occupancy-map precision as 4×4 with 1×1 as shown in Table I. D1 and D2 denote the point-to-point and point-to-plane quality metrics, respectively, for the geometry. We can see that setting occupancy-map precision as 1×1 can achieve an average of 3.9%, 6.7%, and 5.4% performance improvements compared with 4×4 for the Luma, Cb, and Cr components, respectively. However, for geometry, it leads to 105.6% and 85.3% bitrate increase for D1 and D2, respectively, due to the increase in the size of the occupancy. According to our observation, the bit cost of setting occupancy-map precision as 1×1 is 4 times more than 4×4. For this reason, unoccupied-occupied pixels exist under the V-PCC reference software.

To address problems brought by the unoccupied-occupied pixels as mentioned in Section II, we propose padding them with the real points in the original DPC. The real points can come from two aspects. One is the missed points that do not have corresponding pixels in the original 2D projections. The

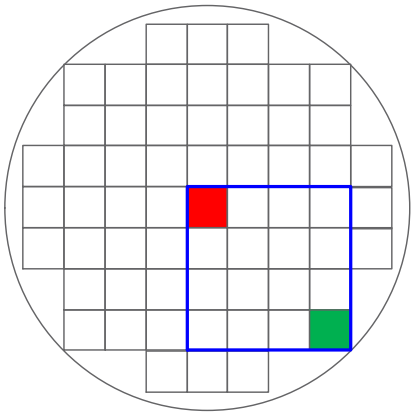


Fig. 5. Search range illustration of the proposed padding method for the unoccupied-occupied pixels. The large blue square represents the 4×4 block to be padded. Each small square represents a pixel.

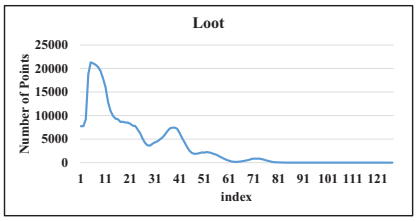


Fig. 6. Histogram of the index of the nearest points selected from the DPC "Loot". The index indicates the distance between the query point and the padded point. The larger the index, the longer the distance.

other is the duplicate points that have corresponding pixels in the other patches. The real points in the original DPC can reduce distortions, especially the point-to-plane errors. However, they may lead to unsmoothness of the padded block, which finally results in a serious bitrate increase. In this work, we aim to find a good balance between the increased bitrate and the reconstructed quality of the DPC.

Before padding of the unoccupied-occupied pixels, we should first locate the to-be-padded 4×4 blocks. We process all the 4×4 blocks one after another and find those with both occupied and unoccupied pixels. While locating those 4×4 blocks, we simultaneously record the 3D coordinate of the first occupied pixel $(x3_q, y3_q, z3_q)$ in the block. This pixel is used as the query point when searching for the to-be-padded candidates in 3D space.

After the to-be-padded 4×4 blocks are located, we then attempt to find the candidate points to be padded. As the final padded point and the query point are in the same 4×4 block, the 2D distance between them in the geometry frame is very small. Since their depth values should also be similar to retain the smoothness of the 4×4 block, their 3D distance should also be small. Therefore, we find several nearest points of the query point $(x3_q, y3_q, z3_q)$ as the candidate points. In our implementation, we use the kd-tree [28] built for all the points in the V-PCC reference software to find those nearest points.

We consider an extreme case to determine the number of nearest points K to be searched as shown in Fig. 5. Each small square represents a pixel. The large blue square represents the 4×4 block to be padded. The red square is the query point.

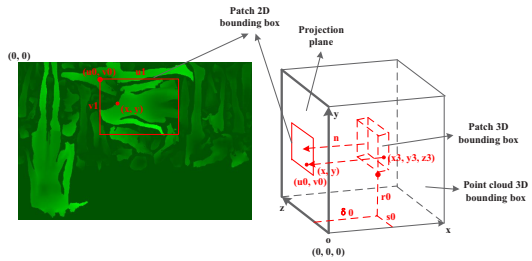


Fig. 7. Illustration of the projection from 3D to 2D coordinates.

The green square is the farthest to-be-padded point compared with the query point. The circle in Fig. 5 includes all the points we need to search in 2D space regardless of the position of the query point. After counting, we need to search 61 points in 2D space. Since a point cloud is essentially the surface of an object, each 2D point will not correspond to more than 2 points in 3D space in most cases. So, we need to search 122 points in 3D space. To avoid some extreme cases where some 2D points may correspond to more than 2 points, we choose to search the nearest 128 points to maximize the RD performance. To better illustrate the reason for setting K as 128, we show the histogram of the index of the nearest points selected of the DPC "Loot" as shown in Fig. 6. The index indicates the distance between the query point and the padded point. The larger the index, the longer the distance. We can see that indices larger than 80 are rarely selected, thus making 128 a good threshold.

As we have explained above, the padded points should satisfy the following constraints. First, they must be real points from the original DPC. To achieve this, the tangential and bi-tangential shifts of the candidate points should be the same as that of the to-be-padded position. This guarantees that the padded points are real points from the original DPC. We give an example to calculate the tangential and bi-tangential shifts of the to-be-padded position as shown in Fig. 7. We assume that the current patch is projected to the yoz plane. Given a to-be-padded position with coordinate (x, y), the start position of the current patch $(u0, v0)$, and the start tangential and bi-tangential shifts of the current patch $(s0, r0)$, the tangential and bi-tangential shifts of the to-be-padded position $(y3, z3)$ can be calculated as

$$\begin{cases} y3 = s0 + (x - u0) \\ z3 = r0 + (y - v0). \end{cases} \tag{1}$$

After deriving the tangential shift and the bi-tangential shift of the to-be-padded position, we cycle through the K candidates to find those satisfying the constraint.

Second, the bitrate of the current block after padding cannot significantly increase. To achieve this, the smoothness of the current block should be kept, which means that the depth difference between the to-be-padded candidates and the occupied pixels should be small. Therefore, we only pad the candidate pixel if the absolute difference between the depth of the to-be-padded pixel $x3$ and the query point $x3_q$ is smaller than a given threshold θ,

$$|x3 - x3_q| < \theta. \tag{2}$$

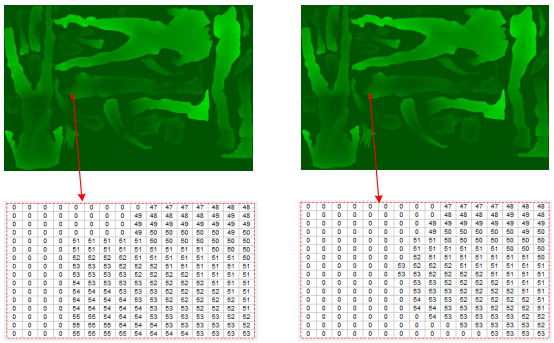


Fig. 8. Comparison between the padded frame using real points (left) and the to-be-padded frame (right). These frames are the first frames of the projected geometry video for the DPC "Loot".

Larger θ values yield more real points in the padded DPC to replace the original unoccupied-occupied pixels, which leads to a higher quality of the reconstructed point cloud. However, the smoothness of the current block will decrease and the bitrate will increase. We set θ to 2.0 as the baseline value, and then analyze the performance of the proposed algorithm for different values of θs in the experimental results.

There may be multiple points satisfying the above two constraints. We choose one based on the projection mode of the current patch. In the V-PCC reference software, each point cloud frame is projected to 2 video frames, d0 and d1. The projection mode determines whether the current patch projects the minimum or the maximum depth to d0 if multiple points have the same tangential and bi-tangential shifts. If the minimum value is projected onto d0, we cycle through the candidate points to find the one with the minimum depth. If the maximum value is projected onto d0, we first find an available candidate point satisfying the two constraints and then cycle through the candidates to find the one with the maximum depth. After finding the most appropriate point $(x3, y3, z3)$, we then project it onto (x, y) using the following formula,

$$\begin{cases} x = z3 - s0 + u0 \\ y = y3 - r0 + v0 \\ h(x,y) = x3 - \delta 0, \end{cases} \tag{3}$$

where $h(x, y)$ is the padded geometry value. Note that we pad both d0 and d1 with $h(x, y)$.

Fig. 8 gives a comparison between the padded frame using real points (left) and the to-be-padded frame (right). As seen in the figure, despite being padded with real points, the block is still smooth. Furthermore, the padding may come from either missed points or duplicate points from other patches. This is verified in the experimental results. Either way, the padded pixels improve the reconstructed quality of the DPC. The above process pads only part of the unoccupied-occupied pixels. As can be seen the pixel in the 1st row and the 9th column of the left figure in Fig 8 is still not padded. Those pixels are then padded using the average value of the 4-neighbor pixels. After using the above steps to generate a video from the original DPC, the resulting video is compressed to finish the encoding process.

We also attempt to pad d1 with a different value from d0 using a similar process. However, due to the bitrate increase, the experimental results suggest a degradation in the overall V-PCC performance. Therefore, we pad d1 with the same value as d0 for the unoccupied-occupied pixels. To clearly delineate the proposed unoccupied-occupied algorithm above, we list the process steps in Algorithm 1.

Algorithm 1 Proposed unoccupied-occupied padding algorithm

Input: Center point $(x3_q, y3_q, z3_q)$, to-be-padded position (x, y), projection mode PM
Output: Padded value $h(x, y)$
Find the nearest K points of $(x3_q, y3_q, z3_q)$
Calculate the tangential and bi-tangential shifts of the candidate points $(y3, z3)$ according to (1)
$x3 = infinitDepth$ ▷ $x3$ initialized as the infinite depth
for $i \leftarrow 1$ **to** K **do**
    if $PM = 0$ and $|x3_i - x3_q| < \theta$ and $x3_i < x3$ **then** ▷ $x3_i$ is the ith candidate
        $x3 = x3_i$
    else if $PM = 1$ **then**
        if $x3 = infinitDepth$ and $|x3_i - x3_q| < \theta$ **then**
            $x3 = x3_i$
        else if $|x3_i - x3_q| < \theta$ and $x3_i > x3$ **then**
            $x3 = x3_i$
        end if
    end if
end for
Calculate $h(x, y)$ according to (3)

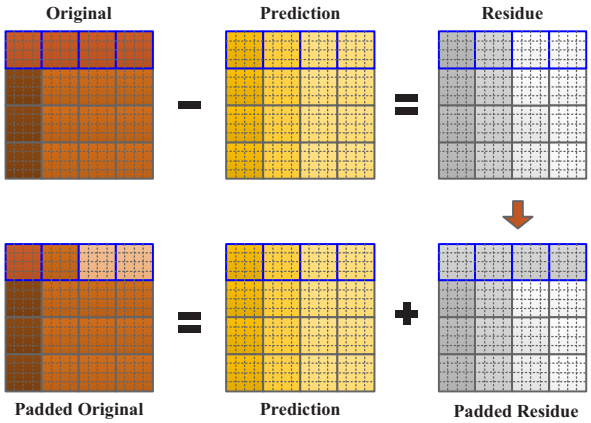


Fig. 9. Basic process of the proposed unoccupied-unoccupied padding. The size of the original block is 16×16. The four small blocks with blue edges are unoccupied 4×4 blocks as indicated by the occupancy map.

B. Padding for unoccupied-unoccupied pixels

As we explained in Section I and Section II, the smoothness of the residual block instead of the original block is the key leading to the lowest bit cost. Therefore, we propose a padding method to smooth the residual block to reduce the bit cost in this section. When we smooth the residual block to reduce the bit cost, we only change the pixel values of the unoccupied 4×4 blocks with only unoccupied-unoccupied pixels according to the occupancy map. We do not change the values of the

occupied 4×4 blocks with both occupied and unoccupied-occupied pixels since they have a significant influence on the reconstructed DPC quality.

Fig. 9 shows the basic process of the proposed padding method for the unoccupied-unoccupied pixels for both the geometry and attribute videos. In the figure, the original block is a PU in HEVC with a size of 16×16. Each small block represents a 4×4 block. The four blocks with blue edges in the original block represent 4 unoccupied 4×4 blocks to be padded. Under the proposed unoccupied-unoccupied padding algorithm, we first obtain a prediction block for the original block from intra or inter prediction in the rate distortion optimization (RDO) process in HEVC. In intra prediction, the prediction block is obtained from the neighboring reconstructed pixels based on the intra prediction direction. In inter prediction, the prediction block is obtained through MC based on the motion information. Then we obtain the residual block through subtracting the prediction block from the original block. After that, the unoccupied-unoccupied pixels of the residual block are padded to minimize the bitrate of the current PU. Finally, the unoccupied-unoccupied pixels of the original block are updated according to the prediction and the padded residue. Using this process, the padding result varies as the prediction varies in the RDO process. Therefore, the proposed unoccupied-unoccupied padding method is a content-adaptive method.

From the above descriptions, we can see that the key step of the proposed unoccupied-unoccupied padding method is correctly padding unoccupied-unoccupied pixels of the residual block to minimize the bit cost of the current PU,

$$\min_{Res_{uo}} R(Res_o, Res_{uo}), \tag{4}$$

where Res_o and Res_{uo} are the sets of the occupied and unoccupied 4×4 blocks of a residual block, respectively. $R(Res_o, Res_{uo})$ is the bitrate of the residual block. Since multiple unoccupied-unoccupied pixels in the current block need to be padded, iterating through all possible combinations of padding pixels to find the optimal solution may significantly increase the encoder complexity. In this work, we propose two intuitive methods to reduce the bitrate.

The first method is to pad unoccupied-unoccupied pixels of the residual block with zeros. This method is similar to the picture boundary padding in [26]. However, the padded zeros combined with the residuals of the occupied 4×4 blocks may increase the variance in the residual block. This makes the residual block difficult to compress.

The second method we propose is to pad unoccupied-unoccupied pixels of the residual block with the average residue of the occupied 4×4 blocks. This method can maintain or increase the smoothness of all residual blocks and is, therefore, more suitable for compression. In the special case where all the 4×4 blocks of the original block are unoccupied, all the unoccupied-unoccupied pixels of the residual block are padded as 0. In this case, the method that pads the average residue of the occupied 4×4 blocks becomes the same as the method that pads 0.

The proposed method is implemented in the RDO process of the HEVC reference software for both intra and inter predictions. The proposed method is implemented in a different way to deal with various kinds of distortions in the RDO process. When the distortion is measured by the sum of the absolute difference (SAD) between the original block and the prediction block in the integer ME process, the SAD is essentially determined by the distortion of the occupied 4×4 blocks regardless of the padding methods used. If we pad the unoccupied-unoccupied pixels of the residual block with 0, the SAD will only be determined by the occupied 4×4 blocks. If we pad the unoccupied-unoccupied pixels of the residual block with the average value of the residues of the occupied 4×4 blocks, the SAD of the whole block will be equal to the SAD of the occupied pixels divided by the ratio of occupied pixels to total pixels. Since this ratio is a fixed value, the SAD of the block is still determined by the occupied 4×4 blocks. To prevent frequently changing the residual and original values of the current block, we add a mask to the current block to calculate the SAD of the occupied 4×4 blocks only,

$$SAD = \sum_{i=1}^{N} |Org_i - Pred_i| \times M_i, \tag{5}$$

where N is the number of pixels in the current block. Org_i and $Pred_i$ are the original and prediction values of pixel i in the current block. M_i is the mask determined by the occupancy map. M_i is 1 when pixel i is occupied and is 0 when pixel i is unoccupied.

When the distortion of the current block is measured using either the sum of the absolute transformed difference (SATD) between the original block and the prediction block or the sum of the squared difference (SSD) between the original block and the reconstructed block, we follow the residual and original padding processes as shown in Fig. 9. Since these methods rely on either a Hadamard transform or discrete cosine transform (DCT), estimating the impact of padding on the distortion after the transform is difficult. Therefore, we have to exactly follow the padding process in Fig. 9. As we limit the search points for calculating the SATD or SSD, the changes in the residual and original values of the current block does not obviously increase the encoder complexity.

In our previous work [29], we proposed an occupancy-map-based RDO to ignore the distortion of unoccupied-unoccupied pixels. In this work, we propose a padding method to reduce the bitrate of blocks containing unoccupied-unoccupied pixels. Both schemes provide significant bitrate savings compared with the V-PCC anchor. Additionally, the unoccupied-unoccupied padding algorithm can have similar effects on the distortions of the unoccupied-unoccupied pixels as the occupancy-map-based RDO. However, combining these two algorithms together leads to further improvement in the RD performance as shown in the experimental results.

IV. Experimental results

A. Simulation setup

The proposed algorithms are implemented in the state-of-the-art video-based point cloud compression (V-PCC) reference software TMC2-4.0 [12] and the corresponding HEVC reference software HM16.18-SCM8.7 [30] to compare with

TABLE II
CHARACTERISTICS OF THE TEST DYNAMIC POINT CLOUDS

Test point clouds	Frame rate (fps)	Points per frame	Geometry precision	Attributes
Loot	30	$\sim$ 780000	10 bit	RGB
RedAndBlack	30	$\sim$ 700000	10 bit	RGB
Soldier	30	$\sim$ 1500000	10 bit	RGB
Queen	50	$\sim$ 1000000	10 bit	RGB
LongDress	30	$\sim$ 800000	10 bit	RGB

TABLE III
PERFORMANCE OF THE COMBINATION OF THE PROPOSED PADDING ALGORITHMS COMPARED WITH THE V-PCC ANCHOR IN THE RA CASE

Test point clouds	Geom.BD-GeomRate D1	D2	Attr.BD-AttrRate Luma	Cb	Cr
Loot	–18.4%	–17.5%	–20.6%	–17.7%	–21.0%
RedAndBlack	–10.6%	–9.7%	–11.9%	–14.9%	–11.6%
Soldier	–16.2%	–15.9%	–11.1%	–5.7%	–5.3%
Queen	–21.3%	–21.5%	–15.7%	–17.8%	–15.7%
LongDress	–11.6%	–10.9%	–4.4%	–4.7%	–4.1%
Avg.	**–15.6%**	**–15.1%**	**–12.7%**	**–12.2%**	**–11.5%**
Enc. self	105%				
Dec. self	99%				
Enc. child	95%				
Dec. child	95%				

TABLE IV
PERFORMANCE OF THE COMBINATION OF THE PROPOSED PADDING ALGORITHMS COMPARED WITH THE V-PCC ANCHOR IN THE AI CASE

Test point clouds	Geom.BD-GeomRate D1	D2	Attr.BD-AttrRate Luma	Cb	Cr
Loot	–9.3%	–8.0%	–9.1%	–1.6%	–5.9%
RedAndBlack	–7.9%	–7.2%	–8.0%	–6.8%	–6.7%
Soldier	–5.1%	–4.5%	–1.8%	6.1%	6.3%
Queen	–8.9%	–9.1%	–8.7%	–9.4%	–8.5%
LongDress	–7.7%	–7.2%	–3.2%	–2.2%	–1.9%
Avg.	**–7.8%**	**–7.2%**	**–6.2%**	**–2.8%**	**–3.4%**
Enc. self	108%				
Dec. self	99%				
Enc. child	100%				
Dec. child	100%				

TABLE V
PERFORMANCE OF THE COMBINATION OF THE PROPOSED PADDING ALGORITHMS IN THE RA CASE COMPARED WITH THE V-PCC ANCHOR TESTED WITH THE FIRST 32 FRAMES

Test point clouds	Geom.BD-GeomRate D1	D2	Attr.BD-AttrRate Luma	Cb	Cr
Loot	–24.2%	–23.5%	–25.7%	–19.0%	–27.1%
RedAndBlack	–10.9%	–9.6%	–12.0%	–14.8%	–11.9%
Soldier	–19.6%	–19.4%	–15.2%	–10.1%	–7.7%
Queen	–18.5%	–18.7%	–15.6%	–15.3%	–11.8%
LongDress	–11.4%	–10.7%	–5.5%	–6.3%	–5.7%
Avg.	**–16.9%**	**–16.4%**	**–14.8%**	**–13.1%**	**–12.8%**
Enc. self	106%				
Dec. self	99%				
Enc. child	95%				
Dec. child	95%				

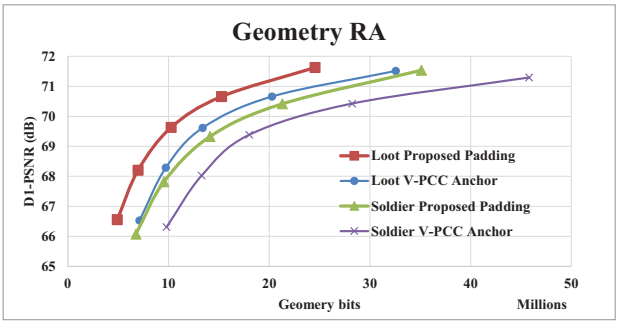
the V-PCC anchor and the state-of-the-art padding algorithms to demonstrate their effectiveness. We test the lossy geometry, lossy attributes, random access (RA) and all intra (AI) cases to demonstrate the effectiveness of the proposed algorithms. We perform experiments on five DPCs defined in the V-PCC common test condition (CTC) [31]. We test all frames of the DPCs for the combination of the proposed padding algorithms. When individually measuring the performance of each algorithm individually, we test on the first 32 frames of the DPCs. The detailed characteristics of the DPCs are shown in Table II. We test all five rate points from low bitrate (r1) to high bitrate (r5) [31]. Since the bitrates generated by the anchor and the proposed algorithms are not the same, the Bjontegaard-delta-rate (BD-rate) [32] is used to compare the respective RD performances.

For the geometry, we report the BD-rates for both point-to-point PSNR (D1) and point-to-plane PSNR (D2) [31]. For the attribute, the BD-rates for the Luma, Cb, and Cr components are reported. For the complexity, we make changes to both the V-PCC and HEVC reference software and, therefore, separately report the encoding and decoding time change for both the V-PCC (self) and HEVC (child) reference software, separately. In the following subsections, we first introduce the overall performance of the combination of the proposed padding algorithms. Then we individually report the performance and analysis of the proposed algorithms.

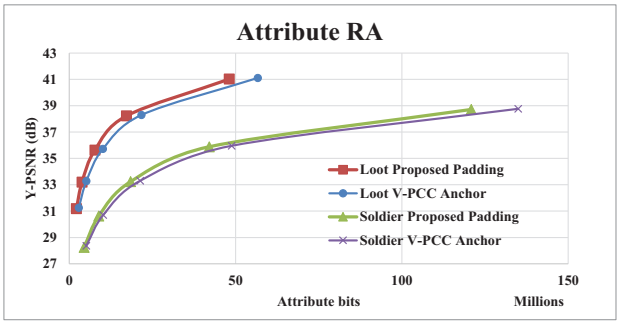
B. Overall performance

Table III and Table IV show the performance of the combination of the proposed algorithms in the RA and AI cases, respectively. The θ is set as 2.0 to obtain a good tradeoff between the reconstructed DPC quality and the bitrate. In Table III, we can see that the combination of the proposed algorithms can achieve an average of 15.6% and 15.1% bitrate savings compared with the V-PCC anchor for the D1 and D2 in the RA case, respectively. The combination of the proposed algorithms can improve the performance by 12.7%, 12.2%, and 11.5% for Luma, Cb, and Cr components, respectively. From Table IV, we can see that the combination of the proposed algorithms can lead to an average of 7.8% and 7.2% bitrate savings compared with the V-PCC anchor for the D1 and D2 in AI case, respectively. The attributes are improved on average by 6.2%, 2.8%, and 3.4% for the Luma, Cb, and Cr components, respectively. We also show some examples of the RD curves in Fig. 10 to better illustrate the performance in both the RA and AI cases. Significant performance improvements are also observed in the RD curves for both the geometry and attribute. From these experimental results, we can conclude that the proposed algorithms significantly improve the performance compared with the V-PCC anchor. We also show that performance improvements when testing the first 32 frames of the DPCs in Table V and Table VI are consistent with the performance of the whole DPC.

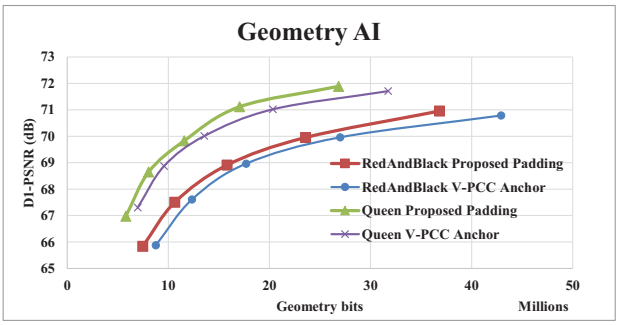
The encoding and decoding complexities of the proposed algorithms are shown in Table III and Table IV, respectively. We can see that the proposed algorithms lead to some complexity increase compared with the V-PCC encoder. This is because of the nearest points search in the unoccupied-occupied padding algorithm in both the RA and AI cases. For the V-PCC decoder, it is unchanged and therefore no complexity change is observed. For the HEVC encoder, the proposed algorithm



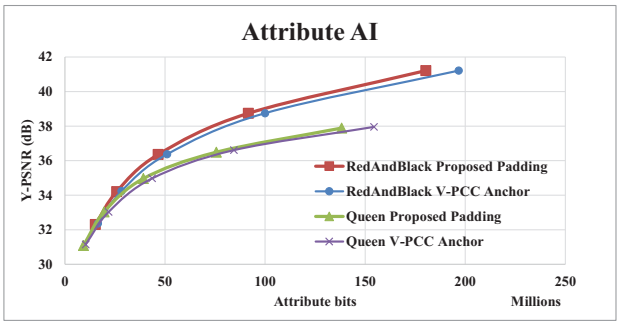
(a) Geometry RA



(b) Attribute RA



(c) Geometry AI



(d) Attribute AI

Fig. 10. Some examples of the RD curves for all the frames of the test DPCs.

TABLE VI
PERFORMANCE OF THE COMBINATION OF THE PROPOSED PADDING ALGORITHMS IN THE AI CASE COMPARED WITH THE V-PCC ANCHOR TESTED WITH THE FIRST 32 FRAMES

Test point clouds	Geom.BD-GeomRate D1	D2	Attr.BD-AttrRate Luma	Cb	Cr
Loot	–9.9%	–8.3%	–8.7%	0.0%	–5.3%
RedAndBlack	–8.3%	–7.2%	–7.5%	–6.5%	–6.7%
Soldier	–6.8%	–6.0%	–2.8%	6.1%	5.9%
Queen	–7.9%	–8.3%	–8.8%	–9.4%	–7.8%
LongDress	–7.7%	–7.3%	–3.4%	–2.2%	–1.9%
Avg.	**–8.1%**	**–7.4%**	**–6.2%**	**–2.4%**	**–3.1%**
Enc. self			108%		
Dec. self			99%		
Enc. child			101%		
Dec. child			100%		

TABLE VII
PERFORMANCE OF THE PADDING OF THE UNOCCUPIED-OCCUPIED PIXELS IN THE RA CASE COMPARED WITH V-PCC ANCHOR TESTED WITH THE FIRST 32 FRAMES

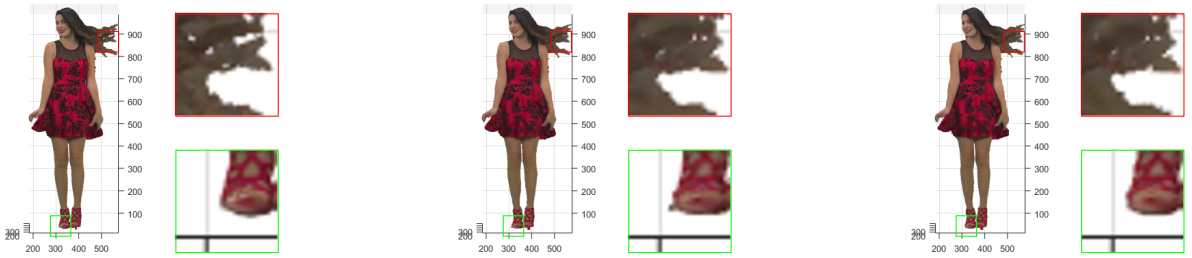
Test point clouds	Geom.BD-GeomRate D1	D2	Attr.BD-AttrRate Luma	Cb	Cr
Loot	–1.4%	–3.9%	0.2%	0.4%	1.2%
RedAndBlack	–1.5%	–4.7%	–0.4%	–0.2%	–0.1%
Soldier	–1.3%	–3.6%	0.6%	3.2%	4.8%
Queen	–4.5%	–4.7%	0.0%	1.5%	1.0%
LongDress	–2.5%	–4.7%	–1.6%	–2.1%	–1.6%
Avg.	**–2.2%**	**–4.3%**	**–0.2%**	**0.6%**	**1.1%**
Enc. self			107%		
Dec. self			100%		
Enc. child			101%		
Dec. child			100%		

decreases the complexity since the unoccupied-unoccupied padding algorithm uses fewer RD operations, especially for the unoccupied blocks in the RA case. For the HEVC decoder, the complexity decreases because it chooses larger blocks for MC in the RA case. In the AI case, the proposed algorithm does not lead to complexity changes for either the HEVC encoder or decoder.

C. Performance of the unoccupied-occupied padding

The performance of the unoccupied-occupied padding method with θ equal to 2.0 in the RA and AI cases are shown in Table VII and Table VIII, respectively. We can see that the padding of the unoccupied-occupied pixels can lead to an average of 2.2% and 4.3% performance improvements for D1 and D2 in the RA case, respectively. The proposed algorithm achieves similar bitrate savings for the geometry in the AI case. However, the proposed algorithm leads to a few performance losses especially for the chroma components since the attribute video becomes less smooth after padding. For the HEVC reference software, the proposed algorithm does not lead to an observable complexity increase for both the encoder and decoder. For the V-PCC reference software, the proposed algorithm may lead to a slight encoder complexity increase caused by finding the nearest neighbors of the current point. We also show one typical example of the subjective quality comparison between the proposed unoccupied-occupied padding and the V-PCC anchor as shown in Fig. 11. Here, we can see that the proposed unoccupied-occupied padding can keep the edges of the hair and shoes better, which leads to a better subjective quality for the DPC.

The reason behind the improvements of the proposed unoccupied-occupied padding method can be seen by com-



(a) Original (b) Anchor (c) Proposed

Fig. 11. Typical example of the subjective quality comparison between the proposed unoccupied-occupied padding and V-PCC anchor. The example is the DPC "RedandBlack" with picture order count 1450 under r1 in the RA case. Note that picture order count 1450 is the first frame of the DPC "RedandBlack" defined in the V-PCC CTC.

TABLE VIII
PERFORMANCE OF THE PADDING OF THE UNOCCUPIED-OCCUPIED PIXELS IN THE AI CASE COMPARED WITH V-PCC ANCHOR TESTED WITH THE FIRST 32 FRAMES

Test point clouds	Geom.BD-GeomRate D1	D2	Attr.BD-AttrRate Luma	Cb	Cr
Loot	–1.3%	–3.9%	0.0%	–0.1%	0.5%
RedAndBlack	–1.7%	–4.7%	0.2%	0.3%	0.4%
Soldier	–1.2%	–4.0%	0.3%	1.2%	1.2%
Queen	–4.4%	–4.8%	1.2%	0.8%	1.3%
LongDress	–1.2%	–3.5%	–0.3%	0.3%	0.4%
Avg.	**–2.0%**	**–4.2%**	**0.3%**	**0.5%**	**0.7%**
Enc. self	107%				
Dec. self	99%				
Enc. child	100%				
Dec. child	100%				

TABLE IX
COMPARISONS OF THE NUMBER OF MISSED POINTS AND DUPLICATE POINTS UNDER THE ANCHOR AND THE PROPOSED UNOCCUPIED-OCCUPIED PADDING IN THE RA CASE FOR THE FIRST 32 FRAMES

Test point clouds	Missed points anchor	proposed	Duplicate points anchor	proposed
Loot	109	106	440461	446985
RedAndBlack	983	931	387439	394802
Soldier	348	330	559069	571041
Queen	4598	4520	306302	315097
LongDress	1323	1258	421661	427674

paring the number of missed points after projection and the number of duplicate points after reconstruction between the V-PCC anchor and the proposed unoccupied-occupied padding as shown in Table IX. Note that the number of duplicate points after reconstruction is the average of all the rate points from r1 to r5. From Table IX, we can see that the number of missed points decreases. This means there is a greater number of real points in the reconstructed DPC, which leads to an improved reconstructed quality. Additionally, the number of duplicate points increases, which indicates that the proposed unoccupied-occupied padding method projects more points in the DPC to multiple patches. When padding an unoccupied-occupied pixel, a duplicate point that has been projected to other patches is a better choice compared with a point that does not have a correspondence in the original DPC. The duplicate points do not decrease the reconstructed DPC quality. These two observations explain the improvements of the proposed algorithm.

TABLE X
PERFORMANCE COMPARISONS OF THE PROPOSED UNOCCUPIED-OCCUPIED PADDING IN THE RA CASE USING DIFFERENT θS FOR THE FIRST 32 FRAMES

θ	Geom.BD-GeomRate D1	D2	Attr.BD-AttrRate Luma	Cb	Cr
1.0	–0.6%	–1.2%	0.1%	0.5%	0.4%
2.0	–2.2%	–4.3%	–0.2%	0.6%	1.1%
3.0	–1.9%	–5.3%	1.5%	3.1%	2.7%

TABLE XI
PERFORMANCE COMPARISONS OF THE PROPOSED UNOCCUPIED-OCCUPIED PADDING IN THE RA CASE USING DIFFERENT NUMBERS OF NEAREST POINTS K FOR THE FIRST 32 FRAMES

K	Geom.BD-GeomRate D1	D2	Attr.BD-AttrRate Luma	Cb	Cr
16	–1.6%	–3.2%	0.0%	0.7%	0.2%
32	–2.0%	–4.0%	–0.1%	1.2%	1.4%
64	–2.2%	–4.3%	–0.1%	0.8%	0.3%
128	–2.2%	–4.3%	–0.2%	0.6%	1.1%

We compare the average performance of the proposed algorithm using different θs in the RA case as shown in Table X. We can see that the proposed algorithm with θ equal to 1.0 leads to geometry compression improvements compared with the V-PCC anchor. However, due to the limited number of padded points, the performance improvement is limited. We can also see from Table X that the proposed algorithm with θ equal to 3.0 can lead to slightly higher performance improvements compared to θ equal to 2.0 for the geometry under the D2 quality measurement. However, this value of θ decreases the smoothness and leads to a bitrate increase for the attribute. Therefore, we choose θ as 2.0 to achieve a better balance between the compression performances of the geometry and attribute.

We also compare the performance of changing the number of nearest points K as shown in Table XI. As we can see, the RD performance keeps improving as the value of

TABLE XII
PERFORMANCE OF THE AVERAGE PADDING OF THE UNOCCUPIED-UNOCCUPIED PIXELS COMPARED WITH V-PCC ANCHOR IN THE RA CASE TESTED WITH THE FIRST 32 FRAMES

Test point clouds	Geom.BD-GeomRate D1	D2	Attr.BD-AttrRate Luma	Cb	Cr
Loot	–20.1%	–19.4%	–26.4%	–21.1%	–25.7%
RedAndBlack	–6.3%	–4.2%	–13.2%	–16.4%	–13.3%
Soldier	–16.0%	–15.6%	–16.8%	–9.1%	–9.1%
Queen	–14.9%	–15.1%	–16.7%	–15.4%	–12.7%
LongDress	–7.2%	–6.1%	–5.2%	–5.9%	–5.4%
Avg.	**–13.0%**	**–12.1%**	**–15.7%**	**–13.6%**	**–13.2%**
Enc. self			101%		
Dec. self			99%		
Enc. child			95%		
Dec. child			95%		

TABLE XIII
PERFORMANCE OF THE AVERAGE PADDING OF THE UNOCCUPIED-UNOCCUPIED PIXELS COMPARED WITH V-PCC ANCHOR IN THE AI CASE TESTED WITH THE FIRST 32 FRAMES

Test point clouds	Geom.BD-GeomRate D1	D2	Attr.BD-AttrRate Luma	Cb	Cr
Loot	–6.4%	–4.1%	–8.9%	0.8%	–4.7%
RedAndBlack	–4.0%	–2.0%	–7.7%	–7.4%	–7.2%
Soldier	–2.9%	–1.6%	–2.9%	5.5%	6.0%
Queen	–3.0%	–3.3%	–9.2%	–9.1%	–7.9%
LongDress	–3.9%	–3.0%	–3.1%	–2.4%	–2.1%
Avg.	**–4.1%**	**–2.8%**	**–6.4%**	**–2.5%**	**–3.2%**
Enc. self			101%		
Dec. self			101%		
Enc. child			102%		
Dec. child			100%		

K increases. Simultaneously, the improvement speed keeps decreasing. Therefore, 128 is a suitable threshold for the proposed unoccupied-occupied padding algorithm.

D. Performance of the unoccupied-unoccupied padding

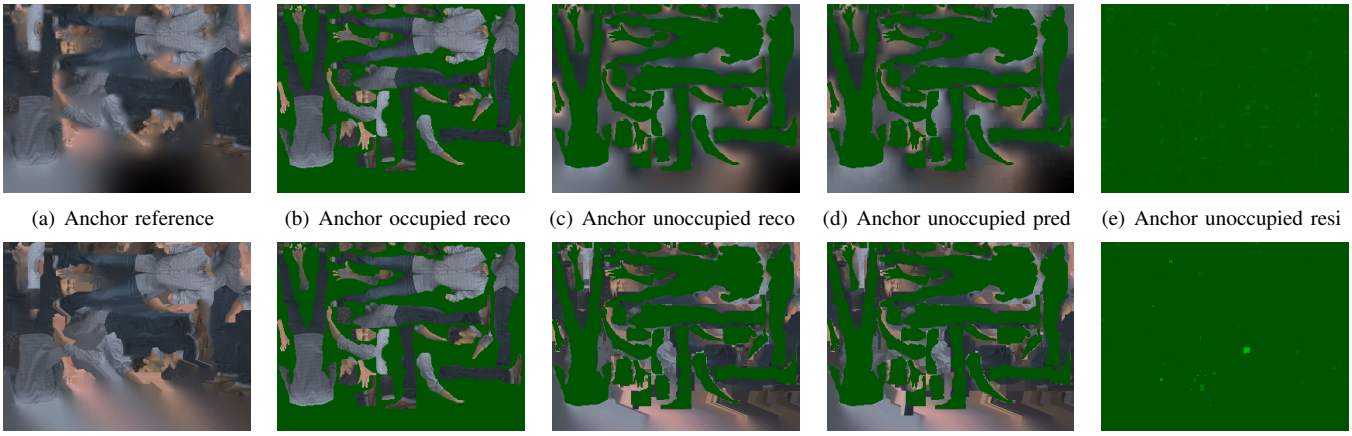
Table XII and Table XIII show the performance of the proposed unoccupied-unoccupied padding in the RA and AI cases, respectively. In the RA case, we can see that the proposed algorithm can achieve an average of 13.0% and 12.1% bitrate savings for the D1 and D2 compared to the V-PCC anchor. For the attribute, the proposed algorithm can bring an average of 15.7%, 13.6%, and 13.2% performance improvements for the Luma, Cb, and Cr components, respectively. In the AI case, the proposed algorithm achieves an average of 4.1% and 2.8% RD performance improvements for the D1 and D2 compared to the V-PCC anchor. For the attribute, average performance improvements of 6.4%, 2.5%, and 3.2% are observed for the Luma, Cb, and Cr components. The experimental results demonstrate that the proposed unoccupied-unoccupied padding algorithm achieves significant bitrate savings compared with the V-PCC anchor. Additionally, we can see that the proposed unoccupied-unoccupied padding achieves a much better performance improvement in the RA case compared with that in the AI case. In essence, the unoccupied-unoccupied pixels under the original padding method in the V-PCC reference software are a combination of the neighboring occupied pixels. Therefore, in the AI case, we achieve a good prediction through intra prediction in most cases. However, we cannot guarantee the correlations between the current frame and its reference frame in inter prediction in the RA case. Therefore, the influence of the unoccupied-unoccupied pixels in the RA case is larger than that in the AI case.

TABLE XIV
PERFORMANCE OF THE PROPOSED UNOCCUPIED-UNOCCUPIED PADDING COMPARED WITH THE SMOOTHED PUSH-PULL ALGORITHM [15] FOR THE ATTRIBUTE IN THE RA CASE TESTED WITH THE FIRST 32 FRAMES

Test point clouds	Geom.BD-GeomRate D1	D2	Attr.BD-AttrRate Luma	Cb	Cr
Loot	0.0%	0.0%	–21.4%	–16.8%	–22.0%
RedAndBlack	0.0%	0.0%	–10.1%	–14.7%	–10.1%
Soldier	0.0%	0.0%	–13.6%	–7.4%	–7.7%
Queen	0.0%	0.0%	–14.6%	–13.6%	–10.0%
LongDress	0.0%	0.0%	–3.3%	–3.4%	–2.7%
Avg.	**0.0%**	**0.0%**	**–12.6%**	**–11.2%**	**–10.5%**
Enc. self			100%		
Dec. self			99%		
Enc. child			95%		
Dec. child			95%		

To better explain the performance of the proposed unoccupied-unoccupied padding, we compare the reference frame, the occupied reconstruction, the unoccupied reconstruction, the unoccupied prediction, the unoccupied residue of the proposed unoccupied-unoccupied padding with the V-PCC anchor as shown in Fig. 12. We can see from Fig. 12 (b) and (g) that the occupied reconstruction of the V-PCC anchor and the proposed unoccupied-unoccupied padding are almost the same. This explains that the proposed unoccupied-unoccupied method will not influence the quality of the reconstructed point cloud. In addition, we can see from Fig. 12 (c) and (h) that the unoccupied reconstruction of the V-PCC anchor and the proposed unoccupied-unoccupied padding are totally different. We can see from Fig. 12 (f) and (i) that most prediction values of the unoccupied-unoccupied pixels under the proposed unoccupied-unoccupied padding method are from co-located positions of the reference frames. Thus, the bit cost for the header information, such as motion vector, is small. We can also see from Fig. 12 (e) and (j) that the residual values of the unoccupied-unoccupied pixels under the proposed unoccupied-unoccupied padding method are much smoother than the anchor. Therefore, the bit cost for the residual information is also small. These two parts lead to a significant performance improvement in the proposed unoccupied-unoccupied padding.

During the V-PCC standardization process, the smoothed push-pull algorithm was developed to further improve the attribute compression performance. The performance comparison between the proposed unoccupied-unoccupied padding and the smoothed push-pull algorithm in the RA and AI cases are shown in Table XIV and Table XV, respectively. We can see that the proposed unoccupied-unoccupied algorithm achieves an average of 12.6%, 11.2%, and 10.5% RD performance improvements in RA case for Luma, Cb, and Cr components, respectively. In addition, it achieves 4.8%, 0.7%, and 0.9% bitrate savings on average in the AI case



(a) Anchor reference (b) Anchor occupied reco (c) Anchor unoccupied reco (d) Anchor unoccupied pred (e) Anchor unoccupied resi

(f) Proposed reference (g) Proposed occupied reco (h) Proposed unoccupied reco (i) Proposed unoccupied pred (j) Proposed unoccupied resi

Fig. 12. Typical examples of the reference frame, occupied reconstruction, unoccupied reconstruction, unoccupied prediction, unoccupied residue from picture order count 16 of the DPC "Loot" encoded under bitrate r4 for the anchor and the proposed unoccupied-unoccupied padding method, respectively. The reco, pred, and resi indicate reconstruction, prediction, and residue, respectively.

TABLE XV
PERFORMANCE OF THE PROPOSED UNOCCUPIED-UNOCCUPIED PADDING COMPARED WITH THE SMOOTHED PUSH-PULL ALGORITHM [15] FOR THE ATTRIBUTE IN THE AI CASE TESTED WITH THE FIRST 32 FRAMES

Test point clouds	Geom.BD-GeomRate D1	D2	Attr.BD-AttrRate Luma	Cb	Cr
Loot	0.0%	0.0%	–6.8%	3.3%	–1.4%
RedAndBlack	0.0%	0.0%	–5.9%	–5.7%	–4.7%
Soldier	0.0%	0.0%	–1.5%	6.4%	7.4%
Queen	0.0%	0.0%	–7.5%	–6.8%	–5.2%
LongDress	0.0%	0.0%	–2.3%	–0.8%	–0.4%
Avg.	**0.0%**	**0.0%**	**–4.8%**	**–0.7%**	**–0.9%**
Enc. self			101%		
Dec. self			101%		
Enc. child			102%		
Dec. child			100%		

TABLE XVI
PERFORMANCE OF THE AVERAGE PADDING COMPARED WITH THE ZERO PADDING IN THE RA CASE FOR THE FIRST 32 FRAMES

Test point clouds	Geom.BD-GeomRate D1	D2	Attr.BD-AttrRate Luma	Cb	Cr
Loot	–2.9%	–2.8%	–0.1%	–20.2%	–19.8%
RedAndBlack	–3.7%	–3.4%	–1.9%	–19.8%	–2.1%
Soldier	–4.3%	–4.1%	–2.3%	–24.9%	–26.4%
Queen	–6.8%	–6.7%	–2.0%	–16.9%	–13.7%
LongDress	–3.6%	–3.5%	–0.5%	–5.5%	–4.3%
Avg.	**–4.3%**	**–4.1%**	**–1.3%**	**–17.4%**	**–13.3%**
Enc. self			100%		
Dec. self			99%		
Enc. child			102%		
Dec. child			100%		

accordingly. The experimental results show that the proposed unoccupied-unoccupied padding algorithm can significantly improve performance compared with the smoothed push-pull algorithm through explicitly smoothing the unoccupied-unoccupied pixels of the residual block.

We also compare the proposed unoccupied-unoccupied padding with average padding to the proposed algorithm with zero padding in the RA case as shown in Table XVI. Here, we can see that the average padding achieves an average of over 4% bitrate improvements for the geometry. For the attribute, the average padding achieves an average of 1.3%, 17.4%, and 13.3% bitrate savings for the Luma, Cb, and Cr components. The experimental results demonstrate the effectiveness of the average padding compared with the zero padding. In addition, we can see that the average padding can achieve the best performance for the chroma components, the second-best for the geometry, and the worst for the Luma component. This is in essence determined by the smoothness of the content. For chroma, as the smoothest of the three components, the zero padding may lead to serious unsmoothness in the residue, leading to a significant increase in the bitrate.

TABLE XVII
PERFORMANCE OF THE PADDING OF THE UNOCCUPIED-UNOCCUPIED PIXELS COMBINED WITH OCCUPANCY-MAP-BASED RDO IN THE RA CASE FOR THE FIRST 32 FRAMES

Test point clouds	Geom.BD-GeomRate D1	D2	Attr.BD-AttrRate Luma	Cb	Cr
Loot	–22.6%	–21.6%	–28.5%	–22.5%	–25.7%
RedAndBlack	–8.3%	–6.2%	–14.7%	–14.5%	–15.1%
Soldier	–17.7%	–17.3%	–18.3%	–11.4%	–8.4%
Queen	–16.7%	–16.8%	–18.3%	–17.8%	–13.6%
LongDress	–9.4%	–8.3%	–7.1%	–6.3%	–6.4%
Avg.	**–14.9%**	**–14.0%**	**–17.4%**	**–14.5%**	**–13.8%**
Enc. self			100%		
Dec. self			99%		
Enc. child			97%		
Dec. child			95%		

In addition, we present the RD performance improvement when the proposed unoccupied-unoccupied padding is combined with the occupancy-map-based RDO [29] [33] in the RA case in Table XVII. Comparing Table XII and Table XVII, we can see that combining the proposed algorithm with the occupancy-map-based RDO achieves extra 1.9% bitrate savings for the geometry under the quality measurements for both D1 and D2. For the attribute, the RD performance improvements of an extra 1.7%, 0.9%, and 0.6% for Luma, Cb, and Cr components, respectively, are achieved. These ex-

perimental results demonstrate that through explicitly ignoring the distortions of the unoccupied pixels, we can obtain some extra performance improvements.

E. Comparisons of the individual algorithms with their combinations

From the experimental results shown above, we can see that the combination of the proposed algorithms outperforms the two, individually. For the geometry, the combination is better than the sum of the two algorithms, individually. When we pad the unoccupied-occupied pixels with the real points from the original DPC, the currently occupied block has a higher probability of finding a good prediction block in the reference frame compared with the V-PCC reference software. The higher probability results in a higher performance improvement for the geometry. For the attribute, however, the increased variance in the residual block due to the unoccupied-occupied pixels may reduce the coding efficiency. Therefore, the performance of the combination is worse than the sum of the performances of the two algorithms.

V. Conclusion

The state-of-the-art 2D-based dynamic point cloud compression algorithm is the video-based point cloud compression (V-PCC) developed by the Moving Pictures Experts Group (MPEG). In this paper, we first point out that the unoccupied pixels in the V-PCC may lead to inefficiency in video compression. Then we divide the unoccupied pixels into two groups based on whether they are signaled as occupied or unoccupied: the unoccupied-occupied pixels and the unoccupied-unoccupied pixels. We propose padding the unoccupied-occupied pixels using the real points in the original dynamic point cloud to increase the number of projected points and reduce some points that do not belong to the original dynamic point cloud. We propose padding the unoccupied-unoccupied pixels through smoothing the residual block instead of the original block to reduce the bitrate as much as possible. The proposed algorithms are implemented in the V-PCC reference software and the corresponding High Efficiency Video Coding reference software. The experimental results show that the proposed algorithms can achieve approximately 16% bitrate savings for the geometry and attribute compared with the V-PCC anchor. The experimental results demonstrate the effectiveness of the proposed algorithms. In the future, we will further investigate more suitable video compression frameworks for the projected videos from the dynamic point cloud.

References

[1] G. Bruder, F. Steinicke, and A. Nüchter, "Poster: Immersive Point Cloud Virtual Environments," in *2014 IEEE Symposium on 3D User Interfaces (3DUI)*, 2014, pp. 161–162.

[2] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-View 3D Object Detection Network for Autonomous Driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1907–1915.

[3] M.-L. Champel, R. Doré, and N. Mollet, "Key Factors for a High-Quality VR Experience," in *2017 SPIE Optical Engineering and applications*, vol. 10396, 2017.

[4] C. Tulvan, R. Mekuria, Z. Li, and S. Laserre, "Use Cases for Point cloud compression (PCC)," Document ISO/IEC JTC1/SC29/WG11 MPEG2015/N16331, Geneva, CH, Jun. 2016.

[5] E. d'Eon, B. Harrison, T. Myers, and P. A. Chou, "8i Voxelized Full Bodies A Voxelized Point Cloud Dataset," Document ISO/IEC JTC1/SC29/WG11 m40059 and ISO/IEC JTC1/SC29/WG1 M74006, Geneva, Switzerland, Jan. 2017.

[6] D. Thanou, P. A. Chou, and P. Frossard, "Graph-Based Compression of Dynamic 3D Point Cloud Sequences," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1765–1778, Apr. 2016.

[7] R. L. de Queiroz and P. A. Chou, "Motion-Compensated Compression of Dynamic Voxelized Point Clouds," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3886–3895, Aug. 2017.

[8] S. Schwarz, M. M. Hannuksela, V. Fakour-Sevom, N. Sheiki-Pour, V. Malamalvadakital, and A. Aminlou, "Nokias Response to CfP for Point Cloud Compression (Category 2)," Document ISO/IEC JTC1/SC29/WG11 m41779, Macau, China, Oct. 2017.

[9] K. Mammou, A. M. Tourapis, D. Singer, and Y. Su, "Video-based and Hierarchical Approaches Point Cloud Compression," Document ISO/IEC JTC1/SC29/WG11 m41649, Macau, China, Oct. 2017.

[10] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokua, S. Lasserre, Z. Li, J. Llach, K. Mammou, R. Mekuria, O. Nakagami, E. Siahaan, A. Tabatabai, A. M. Tourapis, and V. Zakharchenko, "Emerging MPEG Standards for Point Cloud Compression," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 133–148, Mar. 2019.

[11] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[12] "Point Cloud Compression Category 2 Reference Software, TMC2-4.0," http://mpegx.int-evry.fr/software/MPEG/PCC/TM/mpeg-pcc-tmc2.git, accessed: 2019.

[13] E.-C. Ke, S.-P. Wang, Y.-T. Tsai, C.-C. Lin, C.-L. Lin, Y.-H. Lee, J.-L. Lin, Y.-C. Chang, and C.-C. Ju, "[V-PCC] [New proposal] Patch Expansion for Improving Visual Quality," Document ISO/IEC JTC1/SC29/WG11 m47772, Geneva, CH, Mar. 2019.

[14] D. Graziosi, "[V-PCC] TMC2 Optimal Texture Packing," Document ISO/IEC JTC1/SC29/WG11 m43681, Ljubljana, SI, Jul. 2018.

[15] E. Faramarzi and M. Budagavi, "[V-PCC] [New Proposal] Improved Texture Padding," Document ISO/IEC JTC1/SC29/WG11 m46202, Marrakesh, MA, Jan. 2019.

[16] J. Kammerl, N. Blodow, R. B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach, "Real-Time Compression of Point Cloud Streams," in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 778–785.

[17] R. Mekuria, K. Blom, and P. Cesar, "Design, Implementation, and Evaluation of a Point Cloud Codec for Tele-Immersive Video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 4, pp. 828–842, Apr. 2017.

[18] S. Lasserre, J. Llach, C. Guede, and J. Ricard, "Technicolors Response to the CfP for Point Cloud Compression," Document ISO/IEC JTC1/SC29/WG11 m41822, Macau, China, Oct. 2017.

[19] M. Budagavi, E. Faramarzi, T. Ho, H. Najaf-Zadeh, and I. Sinharoy, "Samsungs Response to CfP for Point Cloud Compression (Category 2)," Document ISO/IEC JTC1/SC29/WG11 m41808, Macau, China, Oct. 2017.

[20] L. He, W. Zhu, and Y. Xu, "Best-Effort Projection based Attribute Compression for 3D Point Cloud," in *2017 23rd Asia-Pacific Conference on Communications (APCC)*, Dec. 2017, pp. 1–6.

[21] M. Preda, "Report on PCC CfP Answers," Document ISO/IEC JTC1/SC29/WG11 w17251, Macau, China, Oct. 2017.

[22] D. Graziosi and A. Tabatabai, "[V-PCC] New Contribution on Geometry Padding," Document ISO/IEC JTC1/SC29/WG11 m47496, Geneva, CH, Mar. 2019.

[23] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The Lumigraph," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '96, 1996, pp. 43–54.

[24] K. Mammou, J. Kim, V. Valentin, F. Robinet, A. Tourapis, and Y. Su, "CE2.12 Related: Sparse Linear Model Based Padding Method for the Texture Images," Document ISO/IEC JTC1/SC29/WG11 m44837, Macau, CH, Oct. 2018.

[25] D. Graziosi, "V-PCC New Proposal (related to CE2.12): Harmonic Background Filling," Document ISO/IEC JTC1/SC29/WG11 m46212, Marrakesh, MA, Jan. 2019.

[26] M. Li, Y. Chang, F. Yang, and S. Wan, "Rate-Distortion Criterion Based Picture Padding for Arbitrary Resolution Video Coding Using H.264/MPEG-4 AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 9, pp. 1233–1241, Sept. 2010.
[27] L. Li, Z. Li, X. Ma, H. Yang, and H. Li, "Advanced Spherical Motion Model and Local Padding for 360 Video Compression," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2342–2356, May 2019.
[28] J. L. Bentley, "Multidimensional Binary Search Trees Used for Associative Searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.
[29] L. Li, Z. Li, S. Liu, and H. Li, "Occupancy-Map-Based Rate Distortion Optimization for Video-Based Point Cloud Compression," in *2019 IEEE International Conference on Image Processing (ICIP)*, Sep. 2019, pp. 3167–3171.
[30] "High Efficiency Video Coding Test Model, HM-16.18+SCM8.7," https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/, accessed: 2019.
[31] S. Schwarz, G. Martin-Cocher, D. Flynn, and M. Budagavi, "Common Test Conditions for Point Cloud Compression," Document ISO/IEC JTC1/SC29/WG11 w17766, Ljubljana, Slovenia, Jul. 2018.
[32] G. Bjontegaard, "Calculation of Average PSNR Differences between RD-Curves," Document VCEG-M33, Austin, Texas, USA, Apr. 2001.
[33] L. Li, Z. Li, S. Liu, and H. Li, "Occupancy-map-based rate distortion optimization and partition for video-based point cloud compression," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2020.

Li Li (M'17) is a research fellow with department of electronic engineering and information science, University of Science and Technology of China (USTC). He received the B.S. and Ph.D. degrees in electronic engineering from USTC, Hefei, Anhui, China, in 2011 and 2016, respectively. He was a visiting assistant professor in University of Missouri-Kansas City from 2016 to 2020.

His research interests include image/video coding and processing. He received the Best 10% Paper Award at the 2016 IEEE Visual Communications and Image Processing (VCIP) and the 2019 IEEE International Conference on Image Processing (ICIP).

Zhu Li (M'02-S'07) is an associated professor with the Dept of CSEE, University of Missouri, Kansas City, USA, directs the NSF I/UCRC Center for Big Learning at UMKC. He received his PhD in Electrical & Computer Engineering from Northwestern University in 2004, and was the AFRL Summer Faculty at the US Air Force Academy, UAV Research Center, 2016, 2017 and 2018. He was Sr. Staff Researcher/Sr. Manager with Samsung Research America's Multimedia Core Standards Research Lab in Dallas, from 2012-2015, Sr. Staff Researcher at FutureWei, from 2010 to 2012, Assistant Professor with the Dept of Computing, The HongKong Polytechnic University from 2008 to 2010, and a Principal Staff Research Engineer with the Multimedia Research Lab (MRL), Motorola Labs, Schaumburg, Illinois, from 2000 to 2008. His research interests include image/video analysis, compression, and communication and associated optimization and machine learning problems.

He has 46 issued or pending patents, 100+ publications in book chapters, journals, conference proceedings and standards contributions in these areas. He is Associate Editor-in-Chief (AEiC) for IEEE Trans on Circuits & System for Video Tech, 2020-, and served and serving as Associated Editor for IEEE Trans on Image Processing (2019-), IEEE Trans on Multimedia (2015-2019), and IEEE Trans on Circuits & System for Video Tech (2016-2019). He received a Best Paper Award from IEEE Int'l Conf on Multimedia & Expo (ICME) at Toronto, 2006, and a Best Paper Award from IEEE Int'l Conf on Image Processing (ICIP) at San Antonio, 2007.

Shan Liu (Senior Member, IEEE) received the B.Eng. degree in electronics engineering from Tsinghua University, the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, respectively.

She is a Tencent Distinguished Scientist and General Manager of Tencent Media Lab. She was formerly Director of Media Technology Division at MediaTek USA. She was also formerly with MERL, Sony and IBM. Dr. Liu has been actively contributing to international standards since the last decade. She has numerous proposed technologies adopted into various standards such as HEVC, VVC, OMAF, DASH, PCC and served co-Editor of HEVC SCC and the emerging VVC. At the same time, technologies and products developed under her leadership have reached 10+ million DAU. Dr. Liu holds more than 150 granted US and global patents and has authored or co-authored more than 80 peer reviewed technical articles. She was in the committee of Industrial Relationship of IEEE Signal Processing Society (2014-2015). She is currently on the Editorial Board of IEEE Transactions on Circuits and Systems for Video Technology (2018-2021) and serves as Vice Chair of IEEE Data Compression Standards Committee (2019-). She also served the VP of Industrial Relations and Development of Asia-Pacific Signal and Information Processing Association (2016-2017) and was named APSIPA Industrial Distinguished Leader in 2018.

Houqiang Li (M'10-S'12) received the B.S., M.Eng., and Ph.D. degrees in electronic engineering from the University of Science and Technology of China, Hefei, China, in 1992, 1997, and 2000, respectively, where he is currently a Professor with the Department of Electronic Engineering and Information Science.

His research interests include video coding and communication, multimedia search, image/video analysis. He has authored and co-authored over 100 papers in journals and conferences. He served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from 2010 to 2013, and has been with the Editorial Board of the Journal of Multimedia since 2009. He was the recipient of the Best Paper Award for Visual Communications and Image Processing (VCIP) in 2012, the recipient of the Best Paper Award for International Conference on Internet Multimedia Computing and Service (ICIMCS) in 2012, the recipient of the Best Paper Award for the International Conference on Mobile and Ubiquitous Multimedia from ACM (ACM MUM) in 2011, and a senior author of the Best Student Paper of the 5th International Mobile Multimedia Communications Conference (MobiMedia) in 2009.