Occupancy-map-based Rate Distortion Optimization and Partition for Video-based Point Cloud Compression

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

***Abstract*—The video-based point cloud compression (V-PCC) is the state-of-the-art dynamic point cloud compression method developed by the Moving Pictures Experts Group (MPEG). It projects the point cloud patch by patch to its bounding box and organizes all the patches into a video to utilize the efficient video coding framework. However, the unoccupied pixels among different patches will lead to inefficiency of the video compression. First, the unoccupied pixels are treated equal to the occupied pixels in the rate distortion optimization process. However, the unoccupied pixels are useless for the reconstructed quality of the point cloud. Second, the edges of the occupied and unoccupied pixels can divide a coding unit into arbitrary shapes. Consequently, they are not well-characterized by the regular partitions supported in the video coding framework. Therefore, we propose using occupancy-map-based rate distortion optimization and partition to deal with these two problems. First, the occupancy map is used as a mask to ignore the distortions of the unoccupied pixels when calculating the rate distortion cost of a specified block. This strategy is applied to both intra prediction, inter prediction, and the sample adaptive offset to boost the performance. Second, we propose an occupancy-map-based partition to divide the occupied pixels from different patches and the unoccupied pixels into different prediction units (PUs). The motions of occupied PUs are then predicted using the auxiliary information. The unoccupied PU is finally padded using the approach that generates the geometry frames. The proposed algorithms are implemented in the V-PCC and High Efficiency Video Coding reference software. The experimental results show that the algorithms can individually contribute significant performance improvements compared with the V-PCC. Additionally, their combination can achieve even more bitrate savings than the sum of the individual algorithms.**

***Index Terms*—Occupancy map, Partition, Point cloud compression, Rate distortion optimization, Video-based point cloud compression**

I. Introduction

A point cloud is a set of 3D points that can be used to represent a 3D surface. Each point contains some specific attributes, such as colors, material reflection, and so on. A point cloud can be applied in many virtual reality scenarios as it can be used to reconstruct a 3D object or scene [1]. For example, the point cloud has the potential for 6 degree-of-freedom (DoF) virtual reality, which can achieve a much better user experience than the 3 DoF [2]. It can be used for 3D immersive telepresence [3] as it can be seamlessly integrated and rendered in 3D virtual worlds enabling a convergence between real and virtual realities. Additionally, the point cloud shows considerable potential in auto driving applications by means of its capability to render 3D scenes [4].

Along with the fast development of the point cloud capture technologies, we now have point clouds with millions of points per frame that can render objects with high resolution. On the one hand, the increase in the number of points is the premise of the widespread use of the point cloud. On the other hand, it introduces many burdens for the transmission and storage of the point cloud. For example, one static point cloud (SPC) can have as many as one million points. If 30 and 24 bits are used to represent the geometry and attribute, the original size of one SPC is approximated as 6Mbytes. For a 30 frames per second (fps) dynamic point cloud (DPC), the bitrate will be as high as 180Mbytes per second, which is too much for both transmission and storage. Therefore, there is an urgent need to efficiently compress the point cloud, especially DPC.

The current methods to compress the DPC can be roughly divided into two groups. The first group is the 3D-based method that compresses the DPC in the 3D domain. The 3D-based method usually compresses the geometry of the first frame using octree [5] or binary tree [6], and the attribute of the first frame using transforms such as the Graph Fourier Transform (GFT) [7] or the Region Adaptive Hierarchical Transform (RAHT) [8]. Then, the following frames are divided into multiple cubes. Each cube is predicted from the previous frame using 3D motion estimation (ME) and motion compensation (MC) [9]. However, various frames of a DPC may have a different number of points, and the points do not have explicit correspondence. Therefore, the 3D ME and MC are rather challenging problems and have not yet been well handled.

The second group is the 2D-based method that projects the point cloud to 2D space and organizes the projected point cloud into videos to compress. Some works on this type propose projecting the point cloud directly to a cube or cylinder and organizing the projected faces into a video [10] [11]. These methods can make the projected video with very high spatial

Manuscript received August 15, 2019; revised December 26, 2019; and accepted January 3, 2020. This work was recommended by Associate Editor Dr. Enrico Magli. This work was partially supported by the NSF I/UCRC on Big Learning and Tencent Media Lab.

L. Li and Z. Li are with the Department of Computer Science and Electrical Engineering, University of Missouri-Kansas City, MO 64110, USA. L. Li is also 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. Professor Zhu Li is the corresponding author (e-mail: lil1@umkc.edu; lizhu@umkc.edu).

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

H. Li is 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 (e-mail: lihq@ustc.edu.cn).

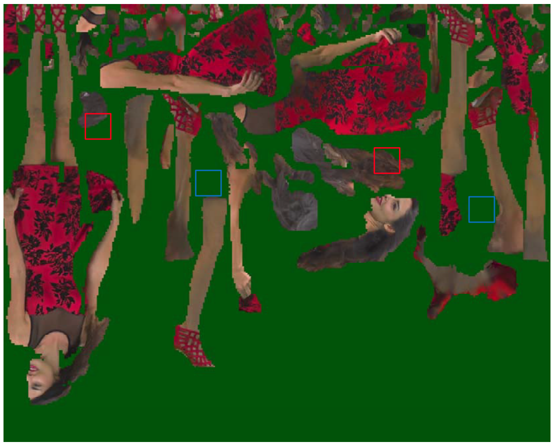


Fig. 1. Typical example of the projected attribute frame of the DPC "RedAndBlack", whose picture order count is 16.

and temporal continuities. However, they may lead to the loss of many points due to occlusion. To solve this problem, the state-of-the-art 2D-based method projects the point cloud patch-by-patch to its bounding box, and organizes all the patches into a video to utilize a video compression framework [12] such as High Efficiency Video Coding (HEVC) [13]. This 2D-based point cloud compression method is the winner of the Moving Picture Experts Group (MPEG) call for proposals for the DPC compression [14]. It is named as the video-based point cloud compression (V-PCC) during the standardization process. After the rise of the V-PCC, many technologies such as auxiliary information coding and patch organization have been proposed to further improve its compression performance [15].

Although the V-PCC achieves much better compression performance compared with the previous DPC compression methods, it produces many unoccupied pixels among various patches to the projected videos, as shown in Fig. 1. We fill all the unoccupied pixels with 0 to better explain the problems. First, the unoccupied pixels are useless for the reconstructed quality of the point cloud as indicated by the blue squares. However, they are treated equal to the occupied pixels in the rate distortion optimization (RDO) process. In particular, under the V-PCC, various patches may be put in different positions in neighboring frames. The unoccupied pixels may cost many bits as they correspond to the occupied pixels in the co-located positions in the reference frame. Second, the edges of the occupied and unoccupied pixels can divide a coding unit (CU) into arbitrary shapes. They cannot be well-characterized by the regular partitions supported in the video coding framework. Even if the binary-tree-quad-tree (BTQT) [16] and triangle partition [17] are considered to be part of the next generation video compression standard [18], they are still unable to solve the problem of the irregular partitions in the patch boundary.

Therefore, in this paper, we propose using the auxiliary information to derive a new rate distortion (RD) criterion and partition to solve these two problems. Since the auxiliary information including the occupancy map and the patch information is encoded before the geometry and attribute videos, it can be used to improve the compression efficiency. The proposed methods mainly have the following contributions.

- The occupancy map is used as a mask when calculating the distortion of a specified block during the RDO process. In this way, the distortion of the unoccupied pixels is ignored. The encoder will choose the best mode to reduce the bit cost of the unoccupied pixels. This strategy is applied to intra prediction, inter prediction, and the sample adaptive offset (SAO) to boost the performance.
- The occupancy map among different patches naturally forms a partition to divide the blocks into occupied prediction units (PUs) from various patches and an unoccupied PU. The occupied PUs are predicted by finding an accurate motion vector predictor (MVP) using the auxiliary information and performing ME based on the MVP. The unoccupied PU is predicted using the approach that generates the geometry frame with each 16×16 block as the basic unit.
- These two technologies are independent of each other and can provide significant performance improvements individually. Since the occupancy-map-based partition can provide a good prediction that is beneficial for the occupancy-map-based RDO process, they can be combined together to contribute more bitrate savings.

The occupancy-map-based RDO has been proposed in our previous work [19]. In this paper, we provide more analysis and experimental results about the occupancy-map-based RDO. Additionally, we propose a completely new occupancy-map-based partition. The experimental results show that the combination of these two technologies can lead to much more performance improvement than the sum of the bitrate savings provided by them individually.

This paper is organized as follows. In Section II, we will provide an overview of the literature of point cloud compression. The proposed occupancy-map-based RDO and partition will be introduced in detail in Section III and Section IV, respectively. Section V will show the detailed experimental results. Section VI will conclude the paper.

II. Related Work

Point cloud compression works can be divided into two groups: SPC compression and DPC compression. Since this paper is more focused on the DPC compression, we will provide a more detailed review on the DPC compression than the SPC compression.

A. Static point cloud compression

The SPC compression algorithms compress the geometry and attribute separately. The most common geometry compression algorithm is the octree-based compression. Botsch *et al.* [20] first proposed dividing the point cloud into 8 subcells in each split. One byte is used to represent whether each subcell is occupied or not. All the bytes are then compressed using arithmetic coding to further improve the compression efficiency. After the octree-based compression was proposed, there were many variations on this topic. For example, Peng *et al.* [21] proposed encoding the number of nonzero subcells N and the combination of the subcells C_8^N to represent each byte. Schnabel and Klein [5] provided a prediction scheme to encode the occupancy of the subcells efficiently. Dricot *et al.* [22] introduced RDO to determine whether a cell should split or not. In addition to the octree-based methods, Chou *et al.* [23] proposed using the blockable geometry to represent the dense point cloud. As some surfaces of the point clouds are planes, Kathariya *et al.* [6] and Park *et al.* [24] proposed using a plane instead of octree to describe the surface. Limuti *et al.* [25] introduced a cellular automata block reversible transform to encode the geometry.

Many works on attribute compression are transform-based methods. Zhang *et al.* [7] proposed using the GFT to utilize the geometry information to exploit the correlations among the attributes. However, the GFT may produce significant complexity burdens to the encoder and decoder as a complex eigenproblem needs to be solved. Therefore, Queiroz and Chou [8] introduced RAHT to compress the attribute to achieve a better balance between the performance and the complexity. In addition to the transform-based methods, a distance-based layered coding structure to compress the attribute from coarse to fine granularities was proposed by Mammou *et al.* [12]. The reconstructed points from the coarse point cloud were used to predict the point cloud with finer granularity. They further proposed using a lifting scheme to improve the performance [26]. Within this topic, some works tried to use the binary tree [27] or some other criteria [28] to determine the points in each layer and improve the coding performance. Additionally, Mekuria *et al.* [29] introduced a snake scanning method to organize the attribute into 8×8 blocks. An image is then constructed by the blocks and compressed using JPEG. Xu *et al.* [30] proposed several scanning modes to convert the attribute into 8×8 blocks and used RDO to select the best one. Nasiri *et al.* [31] proposed an intra prediction method with irregular references and introduced GFT to compact the residue.

B. Dynamic point cloud compression

The DPC compression focuses on how to use ME and MC to exploit the correlations among neighboring frames. As we have mentioned in Section I, the DPC scheme can be divided into the 3D-based method and 2D-based method. For the 3D-based method, Kammerl *et al.* [32] proposed compressing the difference of the octree occupancy to improve the geometry compression efficiency. Daribo *et al.* [33] proposed a curve-based partitioning and exploited the correlations using curve-based intra and inter prediction. However, these methods are inefficient as there are no explicit correspondences among various points in neighboring frames. Thanou *et al.* [34] formulated the 3D ME problem as a feature-matching problem using the spectral graph wavelet descriptors. They considered that the temporal correlations could be better characterized in the spectral domain than in the spatial domain. However, the motion vectors (MVs) of some nodes are inaccurate as they do not have explicit features. Queiroz and Chou [9] attempted to divide each frame into multiple cubes and perform translational ME to determine whether the current block should choose intra or inter prediction. Since the translational motion model is unable to characterize the rotation in DPC, it is difficult for some blocks to find the corresponding blocks. To solve this problem, Mekuria *et al.* [29] proposed using the iterative closest point (ICP) instead of the translational motion model to better exploit the temporal correlations. However, these works can only partially alleviate the problem of 3D ME. They cannot solve the limitation of 3D ME due to the inflexible partitions and inaccurate ME.

For the 2D-based method, the differences of various methods remain in how to generate the video from the DPC. Lasserre *et al.* [35] proposed projecting the points in an octree node to the cube face and organizing all the points into a video. Budagavi *et al.* [36] directly sorted the points in 3D to organize a 2D video. However, these projection methods lead to some shape changes in the 2D video and are unable to utilize the inter prediction technologies in HEVC. To make the generated video easier to be encoded using HEVC, Schwarz *et al.* [10] and He *et al.* [11] proposed projecting the point cloud to a cylinder or cube and organizing the faces into a 2D video. However, these methods will lead to a significant number of missing points due to occlusion. To obtain a better trade-off between the number of projected points and the ability to use the video compression framework, Mammou *et al.* [12] proposed projecting a DPC to a video patch-by-patch and compressing the video using HEVC. This method wins the MPEG call for proposals for the DPC compression. As we have mentioned in Section I, although this method has achieved great success, there is still considerable space to design a better video compression framework utilizing the occupancy map to boost the DPC compression performance. In Section III and Section IV, we will introduce the occupancy-map-based video compression algorithms in detail.

III. Occupancy-map-based rate distortion optimization

In the default encoder of the HEVC reference software (HM), the encoding parameters P are determined by minimizing the following RD cost J,

$$\min_{P} J = \sum_{i=1}^{N} D_i + \lambda R, \tag{1}$$

where N is the number of pixels, D_i is the distortion for a specified pixel i, R is the total bitrate, and λ is the Lagrange multiplier. During the RDO process, the distortion can be the sum of the absolute difference (SAD), the sum of the absolute transformed difference (SATD), or the sum of the squared difference (SSD). Note that the λ is different when various distortion metrics are used. This criterion accumulates the distortions of all the pixels, which means that it treats the distortions of the occupied and unoccupied pixels equally. However, the distortions of the occupied and unoccupied pixels have different influences on the reconstructed quality of the point cloud. Therefore, the current RDO scheme is unsuitable for the projected geometry and attribute videos from DPC.

In this paper, we add an occupancy-map-based mask to the distortion in the RDO scheme to solve this problem. The RD cost of a block after adding the mask is calculated as

$$\min_{P} J = \sum_{i=1}^{N} D_i \times M_i + \lambda R, \tag{2}$$

where M_i is 1 when pixel i is occupied, and M_i is 0 when pixel i is unoccupied. Using this equation, we can ignore the distortions of the unoccupied pixels and put more emphasis on that of the occupied pixels. In the following, we will describe how we fit (2) into the intra prediction, inter prediction, and SAO in HEVC. Note that HEVC is used as an example to explain how to apply (2) to the video compression framework. It can be easily extended to the other video compression standards such as Advanced Video Coding (AVC) [37] or Versatile Video Coding (VVC) [18].

A. Intra prediction

In the HM, the intra mode decision can be roughly divided into 3 steps: rough intra mode decision, precise intra mode decision, and residue quadtree decision. The rough intra mode decision generates several best intra mode candidates using the SATD between the original and prediction signals plus the λ times the bits of the intra prediction direction R_{dir} as the RD cost,

$$\min_{P} J = \sum_{i=1}^{N} SATD_i + \lambda R_{dir}. \tag{3}$$

During this process, the residue bits are not taken into consideration when calculating the R_{dir}. If we add the mask to the SATD calculation, we can find a good prediction for the occupied pixels. However, we may find very bad predictions for the unoccupied pixels that will finally lead to a cost of many bits for the current prediction unit (PU). For the precise mode decision and residue quadtree decision, since the full RDO employing the SSD between the original and reconstruction signals plus the λ times the total bits as the RD cost is used, (2) is applied to the SSD to ignore the distortions of the unoccupied pixels.

B. Inter prediction

In HM, according to different RDO processes, the inter mode can be divided into merge 2Nx2N/skip mode and the other inter modes including inter 2Nx2N and the other partitions. For the merge 2Nx2N/skip mode, as all the merge candidates will go through the full RDO, (2) is applied to ignore the distortions of unoccupied pixels. Note that the occupied pixels of a PU can find a more suitable MV under the merge mode through ignoring the unoccupied pixels even if the PU includes both occupied and unoccupied pixels. However, the merge mode is unable to handle the case where a block contains various parts from multiple patches or the MVs from the merge candidates are inaccurate.

The other inter modes will first go through the integer and fractional ME to find the MV. The SAD and SATD between the original and prediction signals plus the λ times the bits of the motion information R_{motion} is used as the RD cost for the integer and fractional ME processes, respectively. Additionally, for the partitions except for 2Nx2N, the merge mode will calculate the SATD between the original and prediction signals plus the λ times the bits of the motion information R_{motion} to compare with the SATD of the fractional ME. In all these processes, the RD cost is calculated using

$$\min_{P} J = \sum_{i=1}^{N} SAD_i / SATD_i + \lambda R_{motion}. \tag{4}$$

As the residue bitrate is not taken into consideration, we have not applied (2) in the RDO process as explained in Section III-A. After these steps, the occupancy-map-based RDO is used to calculate the RD cost of inter partitions to compare with the other modes in the full RDO process.

C. Sample adaptive offset

The SAO first calculates the average offsets for different types of edge offset (EO) and band offset (BO). Then, the offsets are determined regarding whether they should be added to the reconstructed pixel values of a coding tree unit (CTU) or not using RDO. Therefore, the statistics of the offsets are the key steps to determine the performance of SAO. Under the original RDO process, the statistics of the offsets come from all the pixels in the frame. However, the unoccupied pixels are already encoded with severe distortions under the occupancy-map-based RDO. Therefore, the average offsets will be mostly determined by distortions of the unoccupied pixels.

The offsets derived as above have two disadvantages. First, the occupied pixels will not have suitable offsets and their RD performance will be degraded. Second, we will waste some bits to encode the offsets as they will only be used by the unoccupied pixels. In this work, we choose to only accumulate the differences of the occupied pixels to calculate the offsets to solve these problems. These calculated offsets will be suitable for the occupied pixels while unsuitable for the unoccupied pixels. Therefore, the distortions of the occupied pixels will be reduced while the unoccupied pixels will not choose SAO to avoid wasting bits.

IV. Occupancy-map-based partition

In HEVC, only regular partitions including 2Nx2N, 2NxN, Nx2N, NxN, 2NxnU, 2NxnD, nLx2N, and nRx2N are supported. These regular partitions cannot satisfy the need of arbitrary partitions derived from the occupancy map. Even if the combination of some small regular partitions can characterize the partition derived from the occupancy map, it may lead to a large number of bits spent on signaling the partitions and the corresponding multiple MVs.

A typical example of the occupancy-map-based partition is shown in Fig. 2. Each small square in Fig. 2 represents a 4×4 block. A 4×4 block is the basic unit of the partition since the occupancy map is signaled using a 4×4 block in the V-PCC reference software [38]. We organize all the 4×4 blocks belonging to the same patch as one occupied PU since the 4×4 blocks within the same patch in a local region should have similar motions. All the unoccupied 4×4 blocks are organized as one unoccupied PU that will be predicted after the occupied PUs.

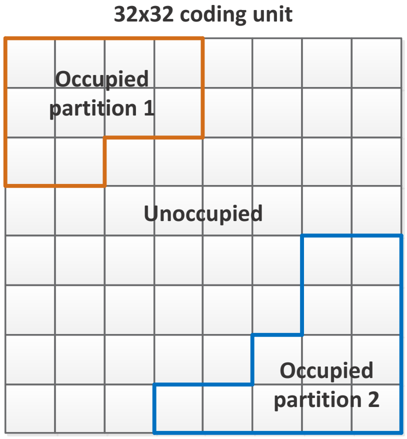


Fig. 2. Typical example of the occupancy-map-based partition of a 32×32 block.

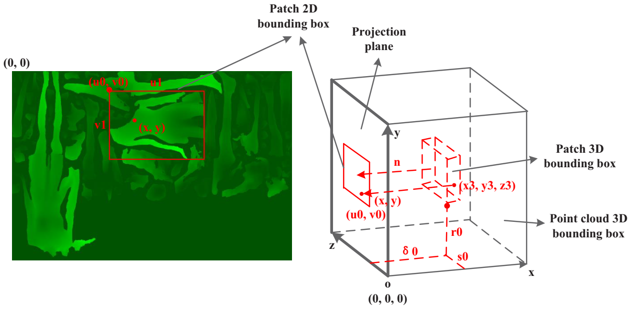


Fig. 3. Illustration of the auxiliary information.

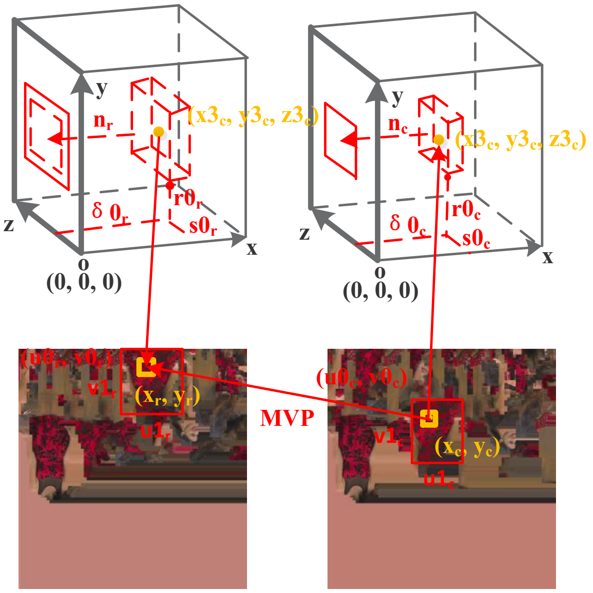


Fig. 4. Auxiliary-information-based motion prediction.

A. *Occupied partition prediction*

Essentially, the prediction of an occupied PU is to find the corresponding block in the reference frame. However, it is difficult to achieve this for the occupied PUs in the patch boundary. In the patch boundary, the spatial correlations between the MV of the occupied PU and its neighboring PUs are very low. Additionally, since the patches with similar content may be packed in different positions in various frames, the temporal correlations between the MVs of the occupied PU and its colocated PUs are also low. Without an accurate MVP from the spatial or temporal neighbors, it is difficult for an occupied PU to find the corresponding block in the reference frame. Even if the occupied PU can find the corresponding block, many bits will be spent on its MV. Both cases will lead to significant bitrate increases and serious point cloud quality degradations.

To handle this problem, we follow the idea in our previous work to provide a better MVP for the occupied PU using the auxiliary information [39] [40]. Before introducing the proposed algorithm, we first provide a brief introduction on the auxiliary information, as shown in Fig. 3. Fig. 3 shows an example of the patch projected to the yoz plane. The auxiliary information includes the index of the projected plane n, the 2D bounding box $(u0, v0, u1, v1)$, and the 3D start location $(\delta 0, s0, r0)$. The $(u0, v0)$ and $(u1, v1)$ are the start position and the size of the 2D bounding box, respectively. To save some bits, the $(u0, v0)$ and $(u1, v1)$ are based on the occupancy resolution OR. The typical value of OR is 16. Based on the auxiliary information, we can derive the 3D coordinate $(y3, z3)$ from the 2D coordinate (x, y) as follows,

$$\begin{cases} y3 = r0 + (y - v0 \times OR) \\ z3 = s0 + (x - u0 \times OR). \end{cases} \tag{5}$$

Note that only two dimensions of the 3D coordinates $(y3, z3)$ can be derived from (x, y) using the auxiliary information. The third dimension, $x3$, is related to the pixel value of the geometry frame.

Fig. 4 illustrates the basic process of the proposed auxiliary-information-based motion prediction. The right part of the figure is the current frame, and the left part is the reference frame. We first find the start coordinate (x_c, y_c) of the first 4×4 block in Z-scan order as the representative of the current PU. We then calculate the current 3D coordinate $(y3_c, z3_c)$ from (x_c, y_c) according to (5),

$$\begin{cases} y3_c = r0_c + (y_c - v0_c \times OR) \\ z3_c = s0_c + (x_c - u0_c \times OR), \end{cases} \tag{6}$$

where $(u0_c, v0_c)$ is the start coordinate of the 2D bounding box of the current patch. $(\delta 0_c, s0_c, r0_c)$ is the start location of the 3D bounding box of the current patch.

After that, we go through all the patches in the reference frame to find the reference patch including $(y3_c, z3_c)$. The reference patch should satisfy the following constraints. First, the reference patch should have the same projection plane index n_r as n_c. If the reference patch has a different projection plane from the current PU, the possibility of the reference patch containing the corresponding block is very low. Even if the reference patch contains the corresponding block, the block with a different projection shape change from the current PU is unsuitable for predicting the current PU. Second, the reference patch should contain $(y3_c, z3_c)$,

$$\begin{cases} s0_r \leq z3_c \leq s0_r + u1_r \times OR - 1 \\ r0_r \leq y3_c \leq r0_r + v1_r \times OR - 1, \end{cases} \tag{7}$$

Algorithm 1 Auxiliary-information-based motion prediction

```
Input: Current PU coordinate $(x_c, y_c)$
Output: Current PU $MVP$
$(u0_c, v0_c)$, $(\delta 0_c, r0_c, s0_c) \leftarrow (x_c, y_c)$
$(z3_c, y3_c) \leftarrow (x_c, y_c)$
$bestDist = MAX$
$bestIndex = 0$        ▷ $bestIndex$ is the best patch index
for $i \leftarrow 1$ to $N_r$ do        ▷ $N_r$ is number of patches
    if $n_r = n_c$ and $s0_r \leq z3_c \leq s0_r + u1_r \times OR - 1$ and $r0_r \leq y3_c \leq r0_r + v1_r \times OR - 1$ then
        $dist \leftarrow abs(\delta 0_c - \delta 0_r)$
        if $dist < bestDist$ then
            $bestDist = dist$
            $bestIndex = i$
        end if
    end if
end for
$(u0_r, v0_r)$, $(r0_r, s0_r) \leftarrow bestIndex$
$MVP \leftarrow (u0_r, v0_r)$, $(r0_r, s0_r)$, $(u0_c, v0_c)$, $(r0_c, s0_c)$
```

where $(\delta 0_r, s0_r, r0_r)$ and $(u0_r, v0_r, u1_r, v1_r)$ are the 3D start location and the 2D bounding box of the reference patch, respectively.

There may be multiple patches satisfying the above two constraints. We choose the one with fewer distances between $\delta 0_c$ and $\delta 0_r$ since the current frame and the reference frame should not change seriously in 3D space. Then, we can find the correspondence between the 3D coordinate $(y3_c, z3_c)$ and the 2D coordinate (x_r, y_r) as

$$\begin{cases} y3_c = r0_r + (y_r - v0_r \times OR) \\ z3_c = s0_r + (x_r - u0_r \times OR). \end{cases} \tag{8}$$

Combining (6) and (8), we have

$$\begin{cases} r0_c + (y_c - v0_c \times OR) = r0_r + (y_r - v0_r \times OR) \\ s0_c + (x_c - u0_c \times OR) = s0_r + (x_r - u0_r \times OR). \end{cases} \tag{9}$$

Therefore, the MVP is derived as

$$\begin{cases} MVP_y = y_r - y_c = r0_c - r0_r + (v0_r - v0_c) \times OR \\ MVP_x = x_r - x_c = s0_c - s0_r + (u0_r - u0_c) \times OR. \end{cases} \tag{10}$$

This MVP is used as the first candidate in the advanced motion vector prediction list. We have not added the MVP in the merge candidate list since it is determined by the patch information and is unable to characterize the local motions. A detailed description of the auxiliary-information-based motion prediction method is shown in Algorithm 1.

After obtaining the MVP, we then perform ME to find the MV of the current PU. During the ME process, when calculating the SAD or SATD of the current PU, only the pixels in the current PU are used. Specifically, for the SATD calculation, the SATD of a PU is the sum of the SATD for each 8×8 block if the PU width and PU height are the multiples of 8 in the HM. However, we can only guarantee that the occupied PU is composed of 4×4 blocks. Therefore, the SATD is modified to be the sum of the SATD for each 4×4 block accordingly. This method to calculate the SATD is also applied to check the merge mode of the current PU. Additionally, we modify the MC process to be performed for each 4×4 block for the Luma component and 2×2 block for the Chroma component. After the RDO process, the motion information including merge flag, merge index, inter direction, MV, MVP index, and MV difference are set for each occupied PU for future reference.

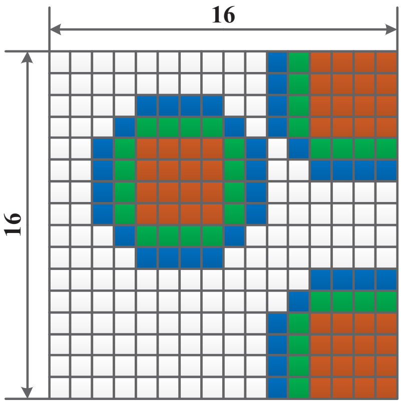


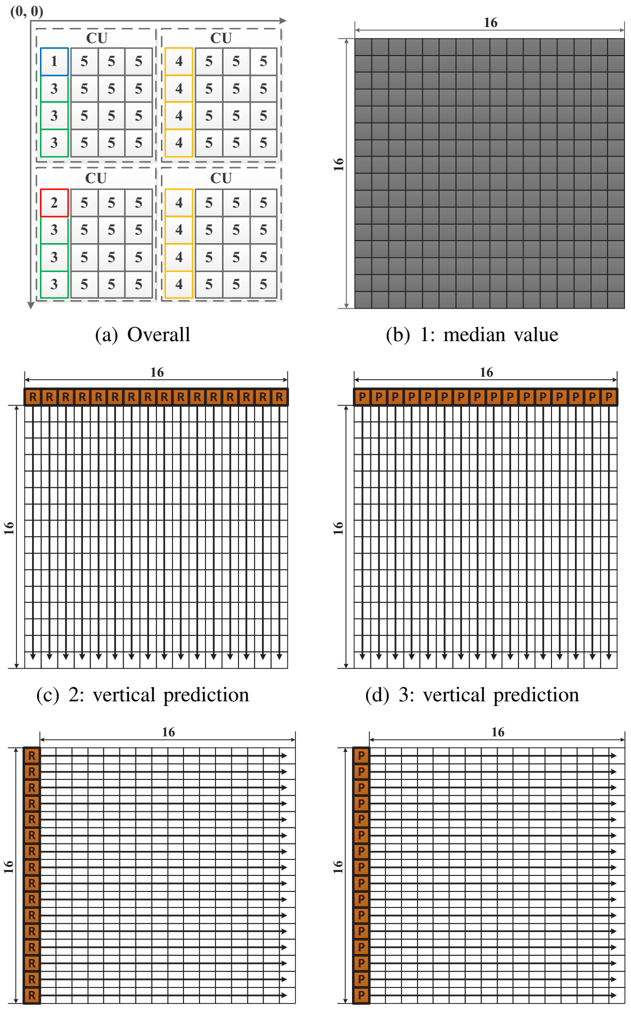
Fig. 5. Padding example of 16 × 16 block with some pixels occupied.

B. Unoccupied partition prediction

After obtaining the predictions of the occupied PUs, we then predict the unoccupied PU of both the geometry and attribute frames in the same way as the padding of the geometry frame. We use the padding method of the geometry frame as it is a local method based on a 16×16 block. The padding of the attribute frame uses a global push-pull algorithm [41] that is difficult to be applied here. The prediction of the unoccupied PU can be divided into two groups according to whether there are occupied pixels in a 16×16 block or not. In the case of an 8×8 CU using the proposed partition, there must be some occupied pixels in the CU, and thus it belongs to the case where there are occupied pixels.

If there are some occupied and unoccupied pixels in a 16×16 block, we will pad the unoccupied pixels according to the occupied ones within the 16×16 block, as shown in Fig. 5. Each small square represents a pixel. The brown pixels are occupied while the other pixels are unoccupied. Note that the occupied pixels are always a group of 4×4 blocks since the occupancy map precision is 4×4. The padding process is a gradual process with multiple iterations. In the first iteration, the green pixels are padded using the average of the 4-neighbor brown pixels. In the second iteration, the blue pixels are padded using the average of the 4-neighbor green pixels. The iterations do not end until all the unoccupied pixels are padded.

If there are only unoccupied pixels in a 16×16 block, the block will be padded using the neighboring reconstructed or predicted pixels. The reconstructed pixels are more accurate than the predicted pixels. The predicted pixels are used when the neighboring pixels have not been reconstructed yet. As shown in Fig. 6 (a), there are five different cases according to the position of the 16×16 block. Each small square represents



(a) Overall

(b) 1: median value

(c) 2: vertical prediction

(d) 3: vertical prediction

(e) 4: horizontal prediction

(f) 5: horizontal prediction

Fig. 6. Five different padding cases of a 16×16 block with only unoccupied pixels. R means reconstruction. P means prediction.

TABLE I
CHARACTERISTICS OF THE TEST DYNAMIC POINT CLOUD

Test point cloud	Frame rate	Number of points	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 II
THE VARIOUS QP SETTINGS UNDER DIFFERENT RATE POINTS

Rate points	Geometry QP	Attribute QP
r1	32	42
r2	28	37
r3	24	32
r4	20	27
r5	16	22

a 16×16 block. As an example, the CU size is the same as the CTU. It can be easily extended to be other CU sizes larger than or equal to 16×16.

- The 16×16 block is the top left 16×16 block of a frame, as shown in position 1 in Fig. 6 (a). In this case, we pad the block using the median value related to the bit depth for the component, as shown in Fig. 6 (b). The median value is 128 for the bit depth 8.
- The 16×16 block is the left boundary of the frame and the top left block of a CU, as shown in position 2 in Fig. 6 (a). In this case, we pad the block using the vertical prediction from the reconstructed pixels of the top boundary, as shown in Fig. 6 (c).
- The 16×16 block is the left boundary of the frame but not the top left block of a CU, as shown in position 3 in Fig. 6 (a). In this case, we pad the block using the vertical prediction from the predicted pixels of the top boundary, as shown in Fig. 6 (d).
- The 16×16 block is the left boundary of the CU but not that of the frame, as shown in position 4 in Fig. 6 (a). In this case, we pad the block using the horizontal prediction from the reconstructed pixels of the left boundary, as shown in Fig. 6 (e).
- The 16×16 block is not the left boundary of the CU, as shown in position 5 in Fig. 6 (a). In this case, we pad the block using the horizontal prediction from the predicted pixels of the left boundary, as shown in Fig. 6 (f).

After the above steps for both the occupied and unoccupied PUs, the prediction block of the current CU is obtained. The prediction block will then go through the full RDO to derive the RD cost and compare with other partitions. When the occupancy-map-based partition is combined with the occupancy-map-based RDO, the occupancy-map-based RDO expressed in (2) is used to further improve the performance. It should also be mentioned that only if there are multiple partitions in the CU, will we signal a flag to the decoder to indicate whether the proposed partition is used. The context model of the flag is the same as that of the merge flag. When the current CU chooses the occupancy-map-based partition, the motion information of the occupied PUs will be signaled to the decoder. We use the same context models as HEVC to signal the motion information.

V. EXPERIMENTAL RESULTS

A. Simulation setup

The proposed algorithms are implemented in the V-PCC reference software TMC2-4.0 [38] and the corresponding HEVC reference software HM16.18-SCM8.7 [42] to compare with the V-PCC anchor to demonstrate their effectiveness. The occupancy-based-RDO can be used for both intra and inter cases, while the occupancy-based-partition is mainly designed for the inter case. Therefore, we test the lossy geometry, lossy attributes, random access case to demonstrate the effectiveness of the proposed algorithm. We test all five DPCs defined in the V-PCC CTC [43] with 32 frames as a good representative of the whole sequence to save some encoding time. The detailed characteristics of all these DPCs are shown in Table I. We test the five rate points from the low bitrate (r1) to the high bitrate (r5) defined in the V-PCC CTC [43]. The settings of the quantization parameters (QPs) of the geometry and attribute videos under different rate points are shown in Table II. Since the bitrates generated by the anchor and the proposed algorithms are not matched exactly, the Bjontegaard Delta rate (BD-rate) [44] is employed to compare the RD performances of various algorithms.

For the geometry, we report the BD-rates for both point-to-point PSNR (D1) and point-to-plane PSNR (D2) [43]. For the attribute, the PSNRs of Luma, Cb, and Cr components are reported. For the complexity of the proposed algorithm, since we have made changes to both the V-PCC and HEVC reference software, we report the encoding and decoding time change for both the V-PCC (self) and HEVC (child), separately. In the following, we will report the performance of the combination of the proposed algorithms followed by that of the proposed algorithms individually. Some analysis of the proposed algorithms, some examples of the RD curves, and a few examples of the subjective quality improvements are also shown to better explain the benefits of the proposed algorithms.

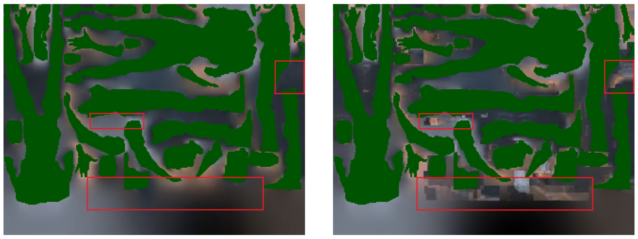
B. Overall performance of the combination of the proposed algorithms

Table III shows the performance of the combination of the occupancy-map-based RDO and partition. Compared with the V-PCC anchor, we can see that the combination of the proposed algorithms can lead to an average of 20.1% and 20.2% bitrate savings for the geometry under the quality metrics D1 and D2, respectively. For the attribute, the combination of the proposed algorithms can achieve on average 36.9%, 39.5%, and 38.4% RD performance improvements for the Luma, Cb, and Cr components, respectively. Taking the total bitrate into consideration, the geometry BD-rate becomes even larger while the attribute BD-rate becomes smaller since the proposed algorithms saved more bitrate for the attribute compared with the geometry. The experimental results demonstrate that the proposed algorithms that make full use of the occupancy map information can lead to very significant bitrate savings compared with the V-PCC anchor.

For the complexities of the V-PCC encoder and decoder, there will be only slight increases since we only output the occupancy map and the auxiliary information. The complexity of the HEVC encoder increases slightly while that of the HEVC decoder decreases slightly since the occupancy-map-based RDO will decrease the complexity while the occupancy-map-based partition will increase the complexity due to the extra added partition. The complexities of the proposed algorithms individually will be introduced later on.

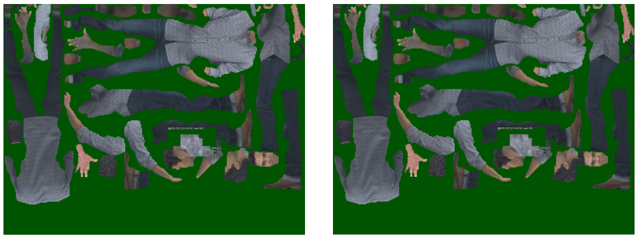
C. Performance of the proposed algorithms individually

1) Occupancy-map-based RDO: Table IV shows the performance of the proposed occupancy-map-based RDO compared with the V-PCC anchor. From Table IV, we can see that the proposed occupancy-map-based RDO can obtain an average of approximately 12% bitrate savings compared with the V-PCC anchor for the geometry. The average performance improvement for the Luma component is 15.4%, which is slightly higher than that of the geometry. The complexity of the HEVC encoder decreases due to the fewer RDO operations since the skip mode can be predetermined for many unoccupied PUs. The complexity of the HEVC decoder also decreases as the decoder uses larger blocks for the unoccupied PUs, which will reduce the complexity of the MC.



(a) Original RDO (b) Occupancy-map-based RDO

Fig. 7. Typical examples of the reconstructed frames with occupied pixels set as 0. The examples are POC 2 of the DPC “Loot” encoded under bitrate r4 using the V-PCC anchor and the occupancy-map-based RDO, respectively.



(a) Original RDO (b) Occupancy-map-based RDO

Fig. 8. Typical examples of the reconstructed frames with unoccupied pixels set as 0. The examples are POC 2 of the DPC “Loot” encoded under bitrate r4 using V-PCC anchor and the occupancy-map-based RDO, respectively.

To better explain the benefits of the proposed occupancy-map-based RDO, we show the comparison of one reconstructed frame from the V-PCC anchor and the occupancy-map-based RDO in Fig. 7 and Fig. 8. Fig. 7 shows the comparison of the unoccupied pixels by setting the occupied pixels as 0. We can see from the red rectangles that the unoccupied pixels are encoded with much worse quality under the occupancy-map-based RDO algorithm compared with the one using the original RDO. We save a number of bits in this way and those unoccupied pixels with bad qualities will not have an influence on the quality of the reconstructed point cloud at all. Fig. 8 shows the comparison of the occupied pixels by setting the unoccupied pixels as 0. We can see that the V-PCC anchor and the proposed occupancy-map-based RDO leads to similar reconstructed qualities for most occupied pixels. In some cases, the occupancy-based RDO achieves better reconstructed quality than the V-PCC anchor, as indicated by the magnified areas since we focus on the occupied pixels to perform RDO. This also contributes to part of the performance improvements.

2) Occupancy-map-based partition: Table V shows the performance of the proposed occupancy-map-based partition compared with the V-PCC anchor. From Table V, we can see that the proposed occupancy-map-based partition can obtain an average of 9.4% bitrate savings for the geometry. For the Luma component, the proposed algorithm can lead to

TABLE III
PERFORMANCE OF THE COMBINATION OF THE OCCUPANCY-MAP-BASED RDO AND PARTITION

Test	Geom.BD-GeomRate		Attr.BD-AttrRate			Geom.BD-TotalRate		Attr.BD-TotalRate		
point cloud	D1	D2	Luma	Cb	Cr	D1	D2	Luma	Cb	Cr
Loot	–25.5%	–25.5%	–47.7%	–52.2%	–53.4%	–30.4%	–30.2%	–34.8%	–38.6%	–39.3%
RedAndBlack	–10.1%	–10.5%	–28.4%	–31.7%	–27.9%	–15.6%	–16.1%	–19.8%	–22.4%	–19.3%
Soldier	–29.8%	–29.7%	–56.0%	–57.5%	–57.8%	–35.8%	–35.7%	–42.7%	–45.9%	–46.2%
Queen	–24.4%	–24.3%	–34.4%	–36.0%	–33.7%	–27.6%	–27.3%	–29.1%	–30.7%	–28.5%
LongDress	–10.8%	–11.0%	–18.1%	–20.1%	–19.1%	–12.4%	–12.9%	–15.3%	–17.1%	–16.2%
Avg.	–20.1%	–20.2%	–36.9%	–39.5%	–38.4%	–24.3%	–24.4%	–28.4%	–30.9%	–29.9%
Enc. time self	101%									
Dec. time self	99%									
Enc. time child	102%									
Dec. time child	95%									

TABLE IV
PERFORMANCE OF THE OCCUPANCY-MAP-BASED RDO

Test	Geom.BD-GeomRate		Attr.BD-AttrRate			Geom.BD-TotalRate		Attr.BD-TotalRate		
point cloud	D1	D2	Luma	Cb	Cr	D1	D2	Luma	Cb	Cr
Loot	–16.3%	–16.4%	–24.3%	–18.2%	–19.3%	–17.8%	–17.9%	–19.5%	–16.1%	–16.5%
RedAndBlack	–6.6%	–7.2%	–12.2%	–9.8%	–12.3%	–8.0%	–8.9%	–9.8%	–8.0%	–9.7%
Soldier	–15.8%	–16.0%	–16.8%	–9.4%	–9.0%	–14.1%	–14.5%	–16.9%	–12.4%	–12.6%
Queen	–13.4%	–13.2%	–15.7%	–11.2%	–10.5%	–13.5%	–13.1%	–14.6%	–12.0%	–11.1%
LongDress	–7.5%	–7.8%	–7.9%	–7.7%	–7.2%	–6.4%	–7.2%	–7.8%	–7.6%	–7.3%
Avg.	–11.9%	–12.1%	–15.4%	–11.3%	–11.7%	–11.9%	–12.3%	–13.7%	–11.2%	–11.4%
Enc. time self	101%									
Dec. time self	99%									
Enc. time child	88%									
Dec. time child	88%									

TABLE V
PERFORMANCE OF THE OCCUPANCY-MAP-BASED PARTITION

Test	Geom.BD-GeomRate		Attr.BD-AttrRate			Geom.BD-TotalRate		Attr.BD-TotalRate		
point cloud	D1	D2	Luma	Cb	Cr	D1	D2	Luma	Cb	Cr
Loot	–10.7%	–10.9%	–15.2%	–22.5%	–25.7%	–11.1%	–11.6%	–12.9%	–17.8%	–19.4%
RedAndBlack	–2.8%	–3.0%	–11.0%	–16.2%	–9.7%	–4.8%	–5.1%	–7.2%	–11.0%	–6.4%
Soldier	–16.8%	–16.8%	–29.4%	–37.3%	–36.0%	–19.5%	–19.5%	–22.9%	–29.4%	–28.6%
Queen	–13.5%	–13.2%	–10.0%	–14.7%	–12.7%	–12.9%	–12.4%	–10.9%	–14.3%	–12.6%
LongDress	–3.1%	–3.1%	–7.4%	–9.6%	–8.8%	–4.7%	–4.7%	–5.8%	–7.5%	–6.9%
Avg.	–9.4%	–9.4%	–14.6%	–20.1%	–18.6%	–10.6%	–10.6%	–11.9%	–16.0%	–14.8%
Enc. time self	101%									
Dec. time self	99%									
Enc. time child	115%									
Dec. time child	105%									

14.6% performance improvement on average. In terms of the complexity, the proposed algorithm will lead to approximately 15% encoder complexity increase due to the extra added partition in the RDO process. The decoder complexity also increases slightly since the 4×4 block MC is used for the CU choosing the occupancy-map-based partition. Compared with the obvious performance improvement achieved by the proposed algorithm, we consider the slight complexity increase as a good trade-off.

To explain the performance source of the proposed algorithms, we disable the padding of the unoccupied PU and use ME to find a prediction for the unoccupied PU. Table VI shows the performance of the proposed algorithm disabling the padding operations compared with that enabling the padding operations. We can see that the performance of disabling the padding operations will lead to 8.8% and 8.9% performance losses for D1 and D2 quality measurements on average, respectively. It will obtain an average of 1.9%, 2.3%, and 1.8% performance improvements for Luma, Cb, and Cr components, respectively. The experimental results demonstrate that the padding method can achieve a good prediction for the unoccupied pixels, especially for the geometry. Additionally, we can see that the padding operations lead to a much better performance for the geometry than that for the attribute. This is because the geometry and attribute use different padding operations in the V-PCC reference software. The geometry is padded using the padding method described in this work. However, the attribute is padded using a global pull-push algorithm that is difficult to be handled for a local block. We apply the padding operations for the geometry to the attribute to find a relatively good prediction and save the bits for the motion information.

To better explain the benefits of the proposed algorithm, Table VII shows the percentage of blocks where the occupancy-

TABLE VI
PERFORMANCE OF THE PROPOSED PARTITION DISABLING THE PADDING OPERATIONS COMPARED WITH THAT ENABLING THE PADDING OPERATIONS

Test	Geom.BD-GeomRate		Attr.BD-AttrRate			Geom.BD-TotalRate		Attr.BD-TotalRate		
point cloud	D1	D2	Luma	Cb	Cr	D1	D2	Luma	Cb	Cr
Loot	10.6%	10.9%	1.9%	1.3%	6.7%	8.8%	9.3%	6.4%	6.1%	8.7%
RedAndBlack	2.7%	2.9%	1.8%	2.3%	1.3%	2.5%	2.8%	2.1%	2.4%	1.8%
Soldier	15.8%	15.9%	2.4%	1.3%	–2.8%	12.3%	12.4%	9.0%	6.8%	4.4%
Queen	12.1%	11.8%	2.2%	5.2%	2.7%	9.6%	9.0%	5.8%	7.8%	6.1%
LongDress	3.0%	3.0%	1.1%	1.2%	1.3%	2.3%	2.4%	1.6%	1.7%	1.8%
Avg.	8.8%	8.9%	1.9%	2.3%	1.8%	7.1%	7.2%	5.0%	5.0%	4.5%
Enc. time self	100%									
Dec. time self	100%									
Enc. time child	102%									
Dec. time child	102%									

TABLE VII
PERCENTAGE OF BLOCKS WHERE THE OCCUPANCY-MAP-BASED PARTITION IS CHOSEN

Test point cloud	Geometry	Attribute
Loot	50.2%	8.1%
RedAndBlack	26.4%	10.1%
Soldier	61.1%	12.4%
Queen	51.8%	8.6%
LongDress	25.7%	9.4%
Avg.	43.0%	9.7%

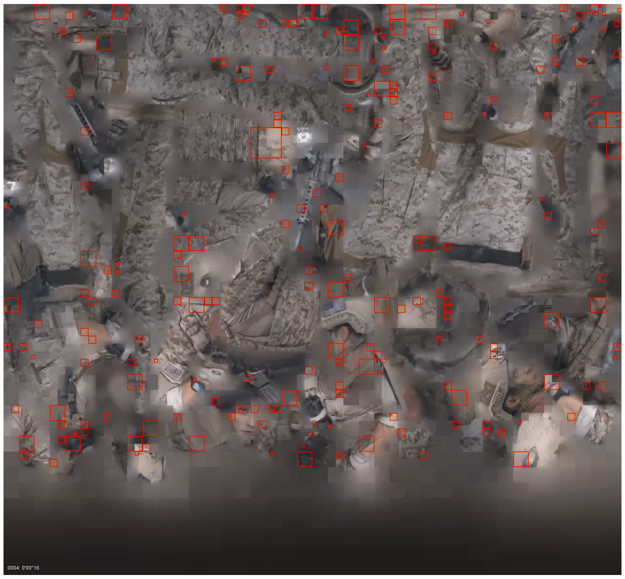


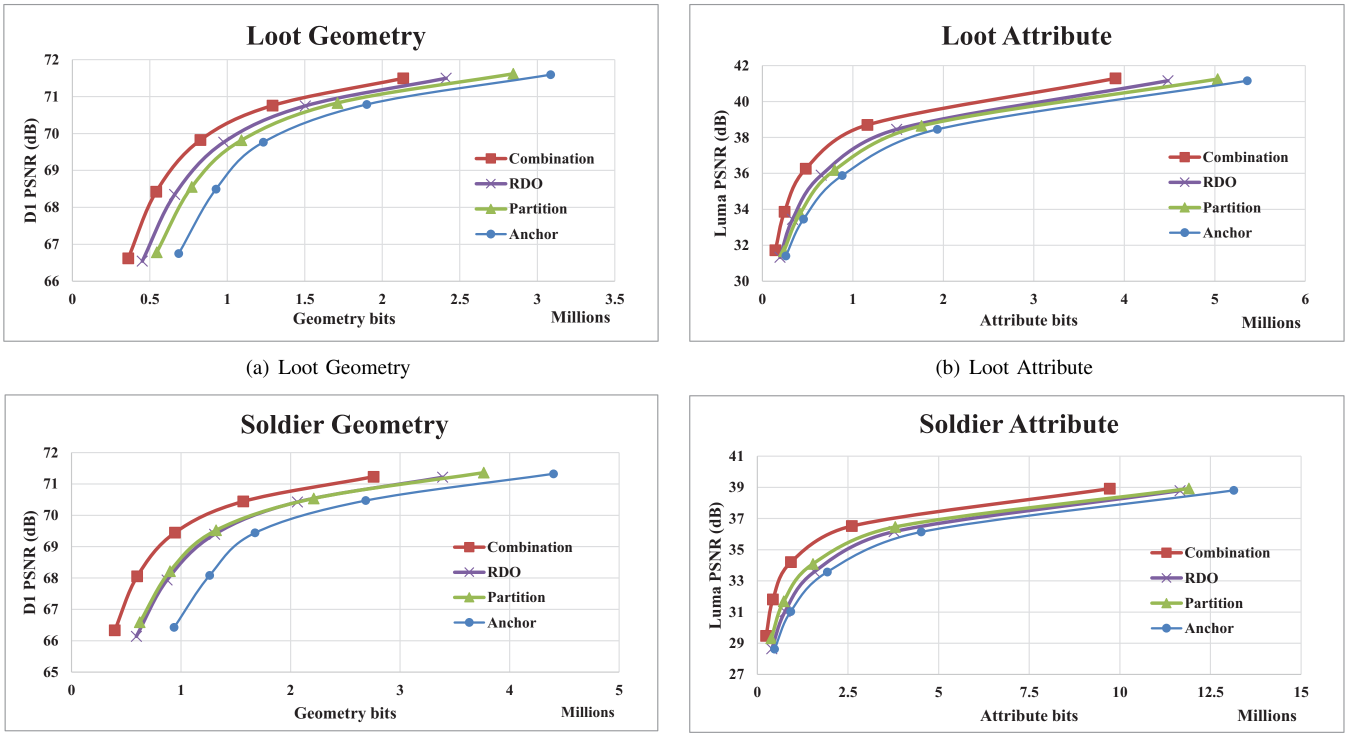
Fig. 9. Typical example of the blocks choosing the occupancy-map-based partition with POC 4 from the DPC "Soldier" encoded under bitrate r4 defined in the CTC.

map-based partition is chosen. The percentage of blocks is calculated using the size of the blocks with the proposed partition dividing the size of the blocks containing both occupied and unoccupied pixels. As we use the padding of the geometry video to predict the unoccupied pixels, the percentage of blocks choosing the occupancy-map-based partition in the geometry video is much more than that in the attribute video. The performance improvement of the geometry video mainly comes from the proposed unoccupied partition prediction algorithm. Therefore, the more the blocks where the proposed occupancy-map-based partition is chosen, the better the performance the test point cloud can achieve. The performance improvement of the attribute video mainly comes from the proposed auxiliary-information-based MVP. The performance improvement is not only determined by the percentage of blocks where the occupancy-map-based partition is chosen, but is also influenced by the propagation of the MVs of the occupied partition. Fig. 9 shows a typical example of the blocks that choose the occupancy-map-based partition. The red squares indicate those blocks choosing the occupancy-map-based partition. We can see that the proposed occupancy-map-based partition obtains good predictions for both the occupied and unoccupied PUs, and therefore improves the performance significantly.

3) Performance comparison with the combination: As seen from Table III, Table IV, and Table V, the sum of the performance improvements of the occupancy-map-based RDO and partition individually is slightly larger than or equal to the bitrate savings of the combination of these two algorithms for the geometry. However, for the attributes, the combination shows clear benefits compared with the sum of the two algorithms. On the one hand, the occupancy-map-based partition can predict the unoccupied pixels well. The occupancy-map-based RDO tries to handle the bit cost of the unoccupied pixels. This may lead to some performance overlaps. On the other hand, only if we can accurately predict the occupied pixels in the patch boundary, will we be able to make full use of the occupancy-based-RDO in the following step by ignoring all the residues. Since the geometry is much smoother than the attribute, the occupied pixels are easier to be accurately predicted even without the proposed algorithms. Therefore, the previous factor is more dominant for the geometry in which some performance overlaps can be observed, while the latter factor is more dominant for the attribute in which clear benefits of the combination algorithm are shown.

D. Typical examples of RD curves

Some examples of the RD curves for both the geometry and attribute are shown in Fig. 10. From Fig. 10, we can see that the proposed occupancy-map-based RDO and partition can achieve very obvious performance improvements compared with the V-PCC anchor. The combination of them can achieve even more bitrate savings than that achieved individually. For the two algorithms individually, for some test DPCs,



(a) Loot Geometry

(b) Loot Attribute

(c) Soldier Geometry

(d) Soldier Attribute

Fig. 10. Typical examples of RD curves.

the occupancy-map-based RDO shows better performance improvements. For the other test DPCs, the occupancy-map-based partition shows slightly better results. Additionally, we can see from the RD curves that the proposed algorithms show more bitrate savings in the low bitrate case compared with that of the high bitrate case.

E. Subjective quality

In addition to the objective quality improvements, the combination of the proposed algorithms also leads to clear subjective quality improvements, as shown in Fig. 11. The left, middle, and right figures are the original point cloud, reconstructed point cloud from the anchor and the combination of the proposed algorithms, respectively. In Fig. 11, we can see that the faces of the woman and soldier become much clearer compared with the anchor even if the bit cost is much lower. The subjective quality improvement can be expected to be much more obvious under the same bitrate. The benefits mainly come from the proposed occupancy-map-based partition that can lead to much better predictions in the low bitrate case. Additionally, the occupancy-map-based RDO that focuses more on the quality of the occupied blocks contributes part of the quality improvements.

VI. Conclusion

In this paper, to address the video compression inefficiency problems that arise from unoccupied pixels, we propose introducing the occupancy-map-based rate distortion optimization and partition into the video coding framework to improve the compression efficiency. The occupancy-map-based RDO mainly addresses the coding units with all the pixels unoccupied. The occupancy-map-based partition tries to handle the coding units with part of the pixels occupied while the other pixels are unoccupied. The proposed algorithms are implemented in the video-based point cloud compression (V-PCC) and the corresponding High Efficiency Video Coding reference software. The experimental results show that the proposed algorithms can individually obtain very obvious performance improvements compared with the V-PCC anchor. The combination of the proposed algorithms can lead to even more bitrate savings. In addition to the objective quality improvements, the proposed algorithm can lead to clear subjective quality improvements. In the future, we will further improve the V-PCC rate distortion performance by designing more efficient video compression tools suitable for V-PCC.

References

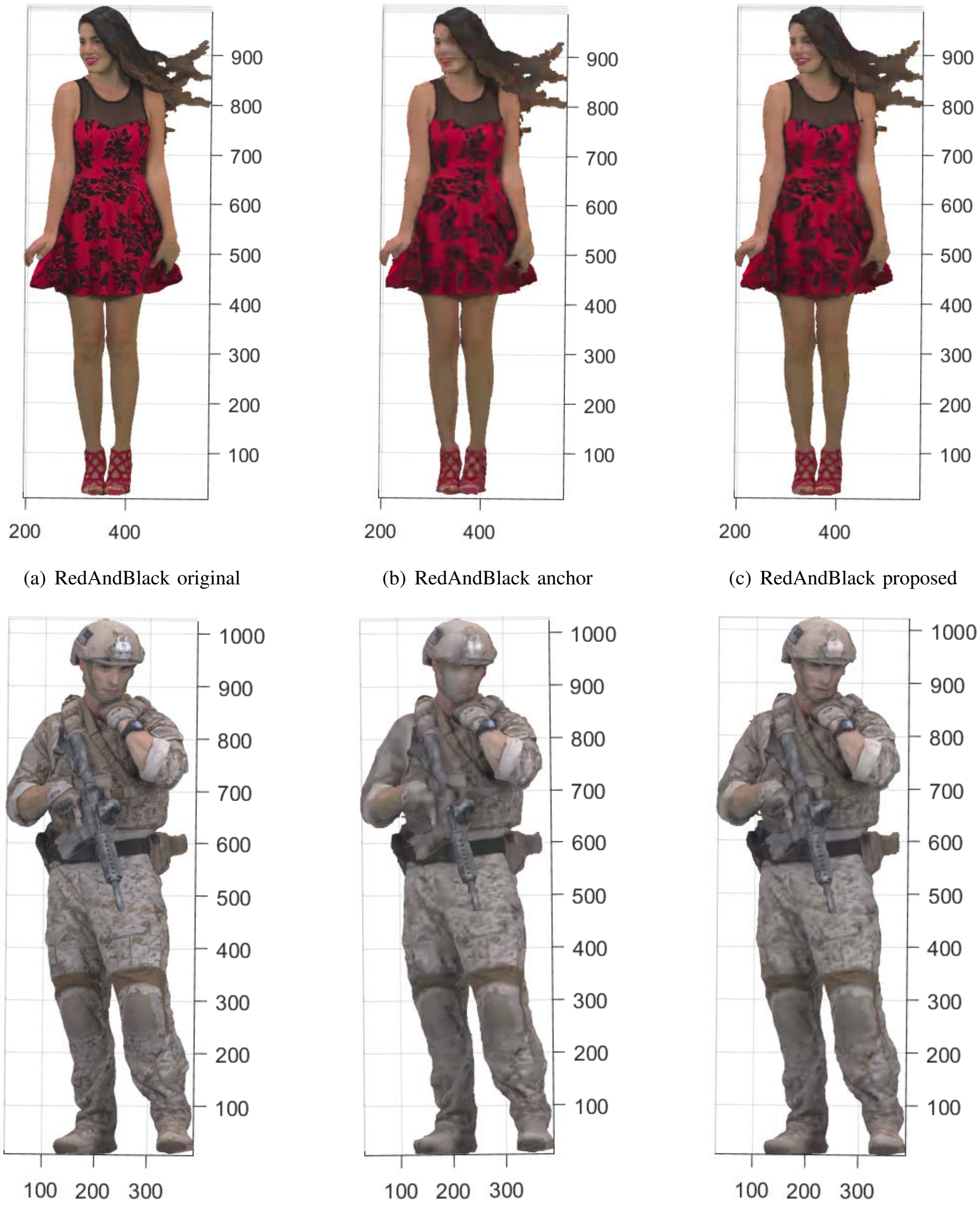
[1] C. Tulvan, R. Mekuria, Z. Li, and S. Laserre, “Use Cases for Point Cloud Compression (PCC),” ISO/IEC JTC1/SC29/WG11 MPEG2015/N16331, Geneva, CH, Jun. 2016.

[2] 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.

[3] H. Fuchs, A. State, and J. Bazin, “Immersive 3D Telepresence,” *Computer*, vol. 47, no. 7, pp. 46–52, Jul. 2014.

[4] 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.

[5] R. Schnabel and R. Klein, “Octree-based Point-Cloud Compression,” *IEEE VGTC conference on Point-Based Graphics*, vol. 6, pp. 111–120, 2006.



(a) RedAndBlack original

(b) RedAndBlack anchor

(c) RedAndBlack proposed

(d) Soldier original

(e) Soldier anchor

(f) Soldier proposed

Fig. 11. Typical examples of the subjective quality comparisons between the anchor and the proposed algorithms. Figures (a)-(c) are from Sequence RedAndBlack, frame 1451. The bit costs of (b) and (c) are 2266848 and 1919088, respectively. Figures (d)-(f) are from Sequence Soldier, frame 537. The bit costs of (e) and (f) are 2302936 and 1529320, respectively.

[6] B. Kathariya, L. Li, Z. Li, J. Alvarez, and J. Chen, "Scalable Point Cloud Geometry Coding with Binary Tree Embedded Quadtree," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*, 2018, pp. 1–6.

[7] C. Zhang, D. Florêncio, and C. Loop, "Point Cloud Attribute Compression with Graph Transform," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct. 2014, pp. 2066–2070.

[8] R. L. de Queiroz and P. A. Chou, "Compression of 3D Point Clouds Using a Region-Adaptive Hierarchical Transform," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3947–3956, Aug. 2016.

[9] ——, "Motion-Compensated Compression of Dynamic Voxelized Point Clouds," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3886–3895, Aug. 2017.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

[15] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou,

R. A. Cohen, M. Krivokuća, 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.

[16] Z. Wang, S. Wang, J. Zhang, S. Wang, and S. Ma, “Effective Quadtree plus Binary Tree Block Partition Decision for Future Video Coding,” in *2017 Data Compression Conference (DCC)*, Apr. 2017, pp. 23–32.

[17] A. Krutz, A. Glantz, T. Sikora, J. Park, S. Park, E. Francois, P. Yin, X. Zheng, H. Yu, W. Han *et al.*, “Tool Experiment 3: Inter Prediction in HEVC,” Document JCTVC-A303, Dresden, Germany, 2010.

[18] B. Bross, J. Chen, and S. Liu, “Working Draft 3 of Versatile Video Coding,” ISO/IEC JTC1/SC29/WG11 N18027, Macao, CN, Oct. 2018.

[19] 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.

[20] M. Botsch, A. Wiratanaya, and L. Kobbelt, “Efficient High Quality Rendering of Point Sampled Geometry,” in *Proceedings of the 13th Eurographics workshop on Rendering*, 2002, pp. 53–64.

[21] J. Peng and C.-C. J. Kuo, “Geometry-Guided Progressive Lossless 3D Mesh Coding with Octree (OT) Decomposition,” *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 609–616, 2005.

[22] A. Dricot, F. Pereira, and J. Ascenso, “Rate-Distortion Driven Adaptive Partitioning for Octree-Based Point Cloud Geometry Coding,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct 2018, pp. 2969–2973.

[23] P. A. Chou, M. Krivokuca, G. Cernigliaro, and E. d’Eon, “Point Cloud Compression Using a Blockable Geometry Representation and Region Adaptive Hierarchical Transform,” ISO/IEC JTC1/SC29/WG11 m41645, Macau, China, Oct. 2017.

[24] S. Park and S. Lee, “Multiscale Representation and Compression of 3-D Point Data,” *IEEE Transactions on Multimedia*, vol. 11, no. 1, pp. 177–183, Jan. 2009.

[25] S. Limuti, E. Polo, and S. Milani, “A Transform Coding Strategy for Voxelized Dynamic Point Clouds,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct. 2018, pp. 2954–2958.

[26] K. Mammou, A. Tourapis, J. Kim, F. Robinet, V. Valentin, and Y. Su, “Lifting Scheme for Lossy Attribute Encoding in TMC1,” Document ISO/IEC JTC1/SC29/WG11 m42640, San Diego, CA, US, Apr. 2018.

[27] V. Zakharchenko, B. Kathariya, and J. Chen, “[G-PCC] [CE13.15] Response on Level of Detail Generation Using Binary Tree for Lifting Transform,” Document ISO/IEC JTC1/SC29/WG11 m45966, Marrakech, MA, Jan. 2019.

[28] K. Mammou, A. Tourapis, and J. Kim, “[G-PCC][New Proposal] Efficient Low-Complexity LOD Generation,” ISO/IEC JTC1/SC29/WG11 m46188, Marrakech, MA, Jan. 2019.

[29] 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.

[30] Y. Xu, S. Wang, X. Zhang, S. Wang, N. Zhang, S. Ma, and W. Gao, “Rate-Distortion Optimized Scan for Point Cloud Color Compression,” in *2017 IEEE Visual Communications and Image Processing (VCIP)*, Dec 2017, pp. 1–4.

[31] F. Nasiri, N. M. Bidgoli, F. Payan, and T. Maugey, “A Geometry-aware Framework for Compressing 3D Mesh Textures,” in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 4015–4019.

[32] 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.

[33] I. Daribo, R. Furukawa, R. Sagawa, H. Kawasaki, S. Hiura, and N. Asada, “Efficient Rate-distortion Compression of Dynamic Point Cloud for Grid-pattern-based 3D Scanning Systems,” *3D Research*, vol. 3, no. 1, p. 2, 2012.

[34] 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.

[35] 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.

[36] 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.

[37] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H. 264/AVC Video Coding Standard,” *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

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

[39] V. Zakharchenko and L. Li, “[V-PCC] [EE2.1 Response] 3D Motion Estimation for Video Compression,” Document ISO/IEC JTC1/SC29/WG11 m45967, Marrakech, MA, Jan. 2019.

[40] L. Li, Z. Li, V. Zakharchenko, J. Chen, and H. Li, “Advanced 3D Motion Prediction for Video-Based Dynamic Point Cloud Compression,” *IEEE Transactions on Image Processing*, vol. 29, pp. 289–302, 2020.

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

[42] High Efficiency Video Coding Test Model, HM-16.18+SCM8.7. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/

[43] 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.

[44] G. Bjontegaard, “Calculation of Average PSNR Differences between RD-curves,” Document VCEG-M33, Austin, Texas, USA, Apr. 2001.

Li Li (M’17) received the B.S. and Ph.D. degrees in electronic engineering from the University of Science and Technology of China (USTC), Hefei, Anhui, China, in 2011 and 2016, respectively. He is now a visiting assistant professor in University of Missouri-Kansas City.

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 is a Tencent Distinguished Scientist and General Manager of Tencent Media Lab. Prior to joining Tencent she was the Chief Scientist and Head of America Media Lab at Futurewei Technologies. She was formerly Director of Multimedia 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 and has numerous proposed technologies adopted into various standards. She holds more than 100 granted US and global patents, some of which have been productized to serve millions of users daily. Dr. Liu served the Industrial Relationship Committee of IEEE Signal Processing Society 2014-2015 and the VP of Industrial Relations and Development of Asia-Pacific Signal and Information Processing Association (APSIPA) 2016-2017. She was named APSIPA Industrial Distinguished Leader in 2018. Dr. Liu obtained her B.Eng. degree in Electronics Engineering from Tsinghua University, M.S. and Ph.D. degrees in Electrical Engineering from University of Southern California.

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.