IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 7, 2021, accepted January 26, 2021, date of publication February 3, 2021, date of current version February 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3056687

Efficient Depth Data Coding Method Based on Plane Modeling for Intra Prediction

DONG-SEOK LEE[1], BYUNG-GYU KIM[2], (Senior Member, IEEE), AND SOON-KAK KWON[1]
[1]Department of Computer Software Engineering, Dong-Eui University, Busan 47340, South Korea
[2]Department of IT Engineering, Sookmyung Women's University, Seoul 04310, South Korea

Corresponding author: Soon-Kak Kwon (skkwon@deu.ac.kr)

This work was supported in part by the BB21+ Project in 2020, and in part by the Ministry of Science and ICT (MSIT), South Korea, through the Grand Information Technology Research Center Support Program, supervised by the Institute for Information and Communications Technology Planning and Evaluation (IITP) under Grant IITP-2021-0-01791.

ABSTRACT In this article, we propose a depth data coding method by plane modeling. The plane modeling is a prediction method based on a plane estimation from depth pixels in a block. The plane modeling improves the intra-picture prediction for depth data comparing with intra modes of conventional coding standards. The plane modeling coefficients, which are the information about the estimated plane, are required to be provided during the depth data coding. The plane modeling coefficients are predicted from neighboring depth pixels. A prediction error of the plane modeling coefficients is calculated through the plane modeling for the selected pixels. If the prediction error is below a certain threshold, then the plane modeling coefficients are applied to the plane modeling for the block. From the simulation results, we verify that the proposed method achieves up to 6.76% bit rate saving in the same coding distortion condition compared to VVC test model.

INDEX TERMS Depth data, RGB-D video, video coding, video compression.

I. INTRODUCTION

Depth data stores the distance information between subjects and camera in pixels. In the certain frame of depth data, the 3D surfaces of the subjects can be reconstructed by a set of the pixels. Depth data enables the following applications using its surface information. The objects can be detected and tracked by detecting specific surfaces from depth data [1]–[3]. Hand gestures can be recognized by tracking a human hand in depth data [4]–[6]. Depth data is also applied to simultaneous localization and mapping by detecting the surfaces of grounds and walls [7], [8]. Transmission and storage may be required for applying depth data for widely fields. For example, a method of face recognition from depth data [9] can be implemented by transmitting and storing depth data between two devices that one device including the depth camera captures a person face, and the other with high-performance processors handles face recognition. An efficient coding method is required for transmission and storage of depth data.

Immersive video, also known as spherical video or 360-degree video, includes depth data for obtaining the distances of subjects in order to restore the video of a specific view. MPEG-I [10], a coding standard for immersive video, codes depth data through the color video coding standard H.265/HEVC [11] or Versatile Video Coding (VVC) [12]. The video coding standard performs intra- and inter-picture predictions for depth data coding. However, the predictions are less accurate for depth data than color video. A correlation between the neighboring pixels in depth data is different from in color video. Therefore, the inter- and intra-picture prediction for depth data should consider the relationship of the adjacent pixels depending on the surface type. The methods for the surface motion estimating between consecutive frames [13] and for calculating the three-dimensional motion of the camera [14], [15] are proposed to predict adjacent pixels by considering the surface type.

The local area of the surface presented by depth data can be approximated as a plane surface. Our previous study [16] proposes a plane modeling method which estimates the approximation plane in blocks for depth data. However, the previous method needs to code the additional coefficients for decoding the estimated plane. The depth data coding by the plane estimation is not improved because of the additional codes.

To solve the problem in the depth data coding of the plane modeling, we propose the depth video coding method

The associate editor coordinating the review of this manuscript and approving it for publication was Gangyi Jiang.

of predicting the plane modeling coefficients. For the depth video encoding by the plane modeling, the plane modeling coefficients are predicted from the neighboring depth pixels. The reference pixels for predicting the plane modeling coefficients are selected in the following order: the depth pixels surrounding the block; the depth pixels in the left box; the depth pixels in the top box. The prediction error of the plane modeling coefficients is calculated through the plane modeling for the selected pixels. If the prediction error is below a certain threshold, then the corresponding plane modeling coefficients are applied to the plane modeling. Otherwise, the next-order reference pixels are selected. If all of the prediction errors are larger than the threshold, then the plane modeling is not applied to the corresponding block. In the decoding process, the depth pixels are reconstructed by restoring the plane modeling coefficients. The reference pixels for the plane modeling coefficient prediction are determined by comparing the prediction errors from already reconstructed pixels.

This article is organized as follows. Conventional coding methods for depth data are described in Section 2. In Section 3, we propose the depth data coding method by the plane modeling. In section 4, we present the simulation results to show the improvement of the depth data coding comparing with VVC, the latest video coding standard. Finally, we will make a conclusion for this article in Section 5.

II. BACKGROUND AND PREVIOUS WORKS

Video can be compressed by removing video redundancy. Video redundancy is classified into a spatial redundancy, a temporal redundancy, and a statistical redundancy. The spatial and the temporal redundancies mean similarities between neighboring pixels in spatial and temporal directions, respectively. The statistical redundancy means repetitive appearances of symbols which represent video. The spatial and temporal redundancies can be reduced by predicting the pixels from neighboring pixels. The statistical redundancy can be reduced by assigning codes of differential length depending on the probability of appearance of the symbols. The color video coding standards, H.264/AVC [17], H.265/HEVC [11], and VVC [12], present various intra-modes, a motion estimation algorithm, and an entropy coding method for removing the spatial, the temporal, and the statistical redundancies, respectively.

Depth data can be reconstructed as three-dimensional surfaces. This implies that adjacent pixels of depth data have a relationship based on a surface type. Therefore, the depth data coding needs to consider the unique relationship of depth data to reduce the redundancy.

Nenci *et al.* [18] propose a compression method by dividing depth data into a multiple channel video. Stankiewicz *et al.* [19] propose a nonlinear transformation in order to coding depth data by extending H.264/AVC. However, these methods do not consider the pixel relationship of depth data based on the surface types. Fu *et al.* [13] proposes an inter-picture prediction method for depth data through surface similarities. The depth pixels in the block are represented as a surface through volumetric integration. The depth pixels are predicted by searching similar surfaces from the neighboring frame of depth data. Wang *et al.* [14], [15] proposed a prediction method for depth data by calculating the camera movement from the neighboring frame. Our previous study [16] predicts the depth pixels through a plane modeling that is an intra-picture prediction method by a plane estimation. The plane is estimated through depth pixel values in the block. The depth pixels are predicted through the estimated plane. The previous study improves prediction accuracies for depth data comparing with the conventional intra modes.

III. PROPOSED DEPTH DATA CODING METHOD FOR INTRA PREDICTION

A. PLANE MODELING FOR DEPTH PICTURE

A depth picture is defined as a frame at the certain time of depth data. A depth pixel value is defined as the z-axis coordinate of the point on 3D space [16]. Therefore, the 3D coordinates of the depth pixel can be calculated through pinhole camera model [20]. Pinhole camera model describes the depth picture as image plane in 3D space whose distance from a depth camera is f, which is a focal length of the camera. Fig. 1 shows that a point on 3D space is projected into image plane in pinhole camera model. 3D coordinates of the point are transformed from image coordinates of the depth pixel as follows:

$$\begin{aligned} X &= \frac{x}{f}p \\ Y &= \frac{y}{f}p \\ Z &= p \end{aligned} \tag{1}$$

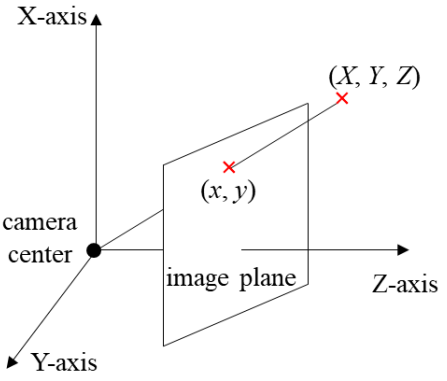


FIGURE 1. Transforming 3D coordinates into 2D image coordinates through pinhole camera model.

where (x, y) and (X, Y, Z) are image and 3D coordinates, respectively, and p is a depth pixel value.

If all of depth pixels in a square block whose height and width are m and n, respectively, are on the same plane in 3D space, then 3D coordinates satisfy a following equation:

$$aX_{ij} + bY_{ij} + c = Z_{ij} \tag{2}$$

where (X_{ij}, Y_{ij}, Z_{ij}) are 3D coordinates of a depth pixel at (i, j) position in the block and a, b, c are the coefficients that determine the plane surface. (2) is expressed in terms of image coordinates by substituting (1) as follows:

$$\alpha x_{ij} + \beta y_{ij} + \gamma = p_{ij}^{-1}, \tag{3}$$

where (x_{ij}, y_{ij}), p_{ij} are image coordinates and a depth pixel value at (i, j) position in the block, respectively, and

$$\begin{aligned} \alpha &= a/(fc) \\ \beta &= b/(fc) \\ \gamma &= -1/c. \end{aligned} \tag{4}$$

If all of depth pixels in a $m \times n$ block are on the same plane, then α, β, and γ satisfy a following equation:

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} \alpha + \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1n} \\ y_{21} & y_{22} & \dots & y_{2n} \\ \dots & \dots & \dots & \dots \\ y_{m1} & y_{m2} & \dots & y_{mn} \end{bmatrix} \beta + \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & 1 \end{bmatrix} \gamma = \begin{bmatrix} p_{11}^{-1} & p_{12}^{-1} & \dots & p_{1n}^{-1} \\ p_{21}^{-1} & p_{22}^{-1} & \dots & p_{2n}^{-1} \\ \dots & \dots & \dots & \dots \\ p_{m1}^{-1} & p_{m2}^{-1} & \dots & p_{mn}^{-1} \end{bmatrix}. \tag{5}$$

If some depth pixels in the block are not on the same plane, then (5) is not satisfied. In this case, errors occur between the left and right sides of (3). The proposed method finds the plane modeling coefficients which minimize the errors. We denote the plane modeling coefficients as PMCs. For finding PMCs, the squares sum of modeling errors is defined as following equation:

$$e^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(\alpha x_{ij} + \beta y_{ij} + \gamma - p_{ij}^{-1} \right)^2. \tag{6}$$

In order to find PMCs, (6) is differentiated into α, β, and γ, respectively, as follows:

$$\begin{aligned} \frac{\partial e^2}{\partial \alpha} &= \sum_{i=1}^{m} \sum_{j=1}^{n} \left(2x_{ij} \left(\alpha x_{ij} + \beta y_{ij} + \gamma - p_{ij}^{-1} \right) \right) = 0 \\ \frac{\partial e^2}{\partial \beta} &= \sum_{i=1}^{m} \sum_{j=1}^{n} \left(2y_{ij} \left(\alpha x_{ij} + \beta y_{ij} + \gamma - p_{ij}^{-1} \right) \right) = 0 \\ \frac{\partial e^2}{\partial \gamma} &= \sum_{i=1}^{m} \sum_{j=1}^{n} \left(2 \left(\alpha x_{ij} + \beta y_{ij} + \gamma - p_{ij}^{-1} \right) \right) = 0. \end{aligned} \tag{7}$$

(7) can be represented as a following matrix equation (8), as shown at the bottom of the page, PMCs, which are the elements of $\mathbf{R}$, are calculated as follows:

$$\begin{aligned} K\alpha = & \sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij} p_{ij}^{-1} \right) \left(mn \sum_{i=1}^{m} \sum_{j=1}^{n} \left(y_{ij}^2 \right) - \left(\sum_{i=1}^{m} \sum_{j=1}^{n} \left(y_{ij} \right) \right)^2 \right) + \sum_{i=1}^{m} \sum_{j=1}^{n} \left(y_{ij} p_{ij}^{-1} \right) \\ & \times \left(\sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij} \right) \sum_{i=1}^{m} \sum_{j=1}^{n} \left(y_{ij} \right) - mn \sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij} y_{ij} \right) \right) + \sum_{i=1}^{m} \sum_{j=1}^{n} \left(p_{ij}^{-1} \right) \\ & \times \left(\sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij} y_{ij} \right) \sum_{i=1}^{m} \sum_{j=1}^{n} \left(y_{ij} \right) - \sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij} \right) \sum_{i=1}^{m} \sum_{j=1}^{n} \left(y_{ij}^2 \right) \right) \\ K\beta = & \sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij} p_{ij}^{-1} \right) \\ & \times \left(\sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij} \right) \sum_{i=1}^{m} \sum_{j=1}^{n} \left(y_{ij} \right) - mn \sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij} y_{ij} \right) \right) \\ & + \sum_{i=1}^{m} \sum_{j=1}^{n} \left(y_{ij} p_{ij}^{-1} \right) \left(\sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij}^2 \right) - \left(\sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij} \right) \right)^2 \right) + \sum_{i=1}^{m} \sum_{j=1}^{n} \left(p_{ij}^{-1} \right) \\ & \times \left(\sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij} \right) \sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij} y_{ij} \right) - \sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij}^2 \right) \sum_{i=1}^{m} \sum_{j=1}^{n} \left(y_{ij} \right) \right) \\ K\gamma = & \sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij} p_{ij}^{-1} \right) \\ & \times \left(\sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij} y_{ij} \right) \sum_{i=1}^{m} \sum_{j=1}^{n} \left(y_{ij} \right) \right. \end{aligned}$$

$$\begin{aligned} \boldsymbol{AR} &= \boldsymbol{B} \\ \boldsymbol{A} &= \begin{bmatrix} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij}^2 \right) & \sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij} y_{ij} \right) & \sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij} \right) \\ \sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij} y_{ij} \right) & \sum_{i=1}^{m} \sum_{j=1}^{n} \left(y_{ij}^2 \right) & \sum_{i=1}^{m} \sum_{j=1}^{n} \left(y_{ij} \right) \\ \sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij} \right) & \sum_{i=1}^{m} \sum_{j=1}^{n} \left(y_{ij} \right) & mn \end{bmatrix} \\ \boldsymbol{R} &= \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \\ \boldsymbol{B} &= \begin{bmatrix} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{ij} p_{ij}^{-1} \right) \\ \sum_{i=1}^{m} \sum_{j=1}^{n} \left(y_{ij} p_{ij}^{-1} \right) \\ \sum_{i=1}^{m} \sum_{j=1}^{n} \left(p_{ij}^{-1} \right) \end{bmatrix}. \end{aligned} \tag{8}$$

$$
\begin{aligned}
&- \sum_{i=1}^{m}\sum_{j=1}^{n}\left(x_{ij}\right)\sum_{i=1}^{m}\sum_{j=1}^{n}\left(y_{ij}^{2}\right)\Big)\\
&+ \sum_{i=1}^{m}\sum_{j=1}^{n}\left(y_{ij}p_{ij}^{-1}\right)\\
&\times\Big(\sum_{i=1}^{m}\sum_{j=1}^{n}\left(x_{ij}\right)\sum_{i=1}^{m}\sum_{j=1}^{n}\left(x_{ij}y_{ij}\right)\\
&- \sum_{i=1}^{m}\sum_{j=1}^{n}\left(x_{ij}^{2}\right)\sum_{i=1}^{m}\sum_{j=1}^{n}\left(y_{ij}\right)\Big)\\
&+ \sum_{i=1}^{m}\sum_{j=1}^{n}\left(p_{ij}^{-1}\right)\\
&\times\Big(\sum_{i=1}^{m}\sum_{j=1}^{n}\left(x_{ij}^{2}\right)\sum_{i=1}^{m}\sum_{j=1}^{n}\left(y_{ij}^{2}\right)\\
&- \Big(\sum_{i=1}^{m}\sum_{j=1}^{n}\left(x_{ij}y_{ij}\right)\Big)^{2}\Big)
\end{aligned}
\tag{9}
$$

where

$$
\begin{aligned}
K = {} & mn\sum_{i=1}^{m}\sum_{j=1}^{n}\left(x_{ij}^{2}\right)\sum_{i=1}^{m}\sum_{j=1}^{n}\left(y_{ij}^{2}\right)\\
&+2\sum_{i=1}^{m}\sum_{j=1}^{n}\left(x_{ij}y_{ij}\right)\sum_{i=1}^{m}\sum_{j=1}^{n}\left(x_{ij}\right)\\
&\times\sum_{i=1}^{m}\sum_{j=1}^{n}\left(y_{ij}\right)\\
&-\sum_{i=1}^{m}\sum_{j=1}^{n}\left(x_{ij}^{2}\right)\Big(\sum_{i=1}^{m}\sum_{j=1}^{n}\left(y_{ij}\right)\Big)^{2}\\
&-\Big(\sum_{i=1}^{m}\sum_{j=1}^{n}\left(x_{ij}\right)\Big)\sum_{i=1}^{m}\sum_{j=1}^{n}\left(y_{ij}^{2}\right)\\
&-mn\Big(\sum_{i=1}^{m}\sum_{j=1}^{n}\left(x_{ij}y_{ij}\right)\Big)^{2}
\end{aligned}
\tag{10}
$$

PMCs of the block are calculated by substituting image coordinates and the depth pixel values into (9).

B. DEPTH DATA CODING WITH PLANE MODELING COEFFICIENTS

The depth pixels in the encoded block by the plane modeling are restored as follows:

$$P_{ij} = \left(\alpha x_{ij} + \beta y_{ij} + \gamma\right)^{-1}. \tag{11}$$

The found PMCs by (9) are required to restore pixels through (11). In order to provide PMCs for restoring the depth pixels, it is available that PMCs are encoded together with the block.

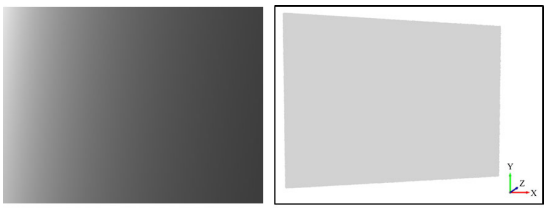
In conventional video coding methods, entropy coding is applied to coding symbols of video. Entropy coding codes a symbol as a variable length code based on the probability of the signal appearance. To code a set of the symbols by entropy coding, the symbols should be finite. However, PMC symbols are infinite because PMC is real number. In order to limit the number of the PMC symbols, PMCs are converted to integer as follows:

$$
\begin{aligned}
\bar{\alpha} &= \lfloor \alpha \times p_{\alpha} \rfloor\\
\bar{\beta} &= \lfloor \beta \times p_{\beta} \rfloor\\
\bar{\gamma} &= \lfloor \gamma \times p_{\gamma} \rfloor,
\end{aligned}
\tag{12}
$$

where p_{α}, p_{β}, and p_{γ} are conversion weights for PMCs and $\lfloor x \rfloor$ is a largest integer less than or equal to x. Entropy coding is applied to the converted PMCs by (12). The plane modeling through the converted PMCs are performed by substituting the $\bar{\alpha}$, $\bar{\beta}$, and $\bar{\gamma}$ into (11) as follows:

$$\bar{P}_{ij} = \left(\bar{\alpha} x_{ij} + \bar{\beta} y_{ij} + \bar{\gamma}\right)^{-1}. \tag{13}$$

The depth data of the plane surface as shown in Fig. 2 is encoded by the plane modeling including PMC encoding for measuring the compression in an ideal situation. Context-adaptive binary arithmetic coding (CABAC) [21] is applied to the PMC encoding. The resolutions of depth data are 640 × 480. The p_{α}, p_{β}, and p_{γ} in (12) are 1, respectively. Depth data is encoded with QP, which is a quantization parameter, in the range of 0∼15. Table 1 shows the improvement rate of bit-rates by applying the plane modeling comparing with VVC. The bit-rate of residuals is reduced up to 10%, but the total bit-rate only decreases up to 2.50%. It means that the bit generation by PMC encoding cancels out a bit reduction by the plane modeling. The selection ratio of the plane modeling is up to only 2.92%. The intra-mode of the plane modeling is rarely selected even in the plane surface area. This result shows that the PMC encoding should be minimized or removed for the depth data compression.



(a) (b)

FIGURE 2. Depth data of plane surface: (a) represented in single channel picture and (b) represented in 3D space.

C. DEPTH DATA CODING BY PLANE MODELING COEFFICIENTS PREDICTION

The PMC encoding reduces the performances of the depth data coding. In order to solve this problem, a PMC prediction method for the depth data coding is proposed. The proposed method is based on that the neighboring depth pixels have a high possibility to be on the same plane. The reference pixels for a PMC prediction are selected from the neighboring depth pixels. We denote the PMC prediction as PMCP. The depth pixels are predicted through the predicted PMCs. The proposed method does not require the PMC encoding, so the depth data encoding can be improved.

Fig. 3 (a)-(c) show the reference pixels which are selected as depth pixels surrounding the block, the depth pixels in a top box, and the depth pixels in a left box, respectively. Gray rectangles in Fig. 3 (a)-(c) indicate the reference pixels. A box height in Fig. 3 (b) is set to within 2∼n and a box width in Fig. 3 (c) is set to within 2∼m. After selecting the reference pixels, PMCs are predicted by a following equation which is modified from (9):

$$\tilde{K}\tilde{\alpha} = \sum_{k=1}^{N}\left(x_k p_k^{-1}\right)\left(N\sum_{k=1}^{N}\left(y_k^{2}\right) - \left(\sum_{k=1}^{N}\left(y_k\right)\right)^{2}\right)$$

TABLE 1. Bit rates of depth data encoding through plane modeling.

QP	Bit-rate of VVC (kbps)	Bit-rate of residuals by applying plane modeling (kbps)	Bit gain of residuals (%)	Bit-rate of PMCs encoding (kbps)	Bit-rate of plane modeling (kbps)	Total bit gain (%)	Selection ratio of plane modeling (%)
0	180.00	161.99	10.00%	16.97	178.96	0.60%	2.73%
1	167.28	151.25	9.60%	15.59	166.85	0.30%	2.90%
2	155.28	140.80	9.30%	13.70	154.49	0.50%	2.86%
3	148.08	136.43	7.90%	11.07	147.50	0.40%	2.77%
4	132.24	120.92	8.60%	10.14	131.06	0.90%	2.64%
5	129.48	118.57	8.40%	9.69	128.26	0.90%	2.96%
6	120.84	109.50	9.40%	10.75	120.25	0.50%	2.71%
7	112.20	102.73	8.40%	8.43	111.16	0.90%	2.81%
8	110.04	101.98	7.30%	7.85	109.84	0.20%	2.92%
9	108.52	102.13	5.90%	5.26	107.39	1.00%	2.19%
10	101.40	95.44	5.90%	4.16	99.61	1.80%	2.39%
11	97.08	91.29	6.00%	3.34	94.62	2.50%	2.26%
12	94.68	90.33	4.60%	3.70	94.02	0.70%	2.15%
13	86.04	81.64	5.10%	3.26	84.90	1.30%	2.26%
14	85.62	81.57	4.70%	3.25	84.82	0.90%	2.47%
15	85.32	81.18	4.90%	2.94	84.12	1.40%	2.08%

$$
\begin{aligned}
&+\sum_{k=1}^{N}\left(y_k p_k^{-1}\right)\left(\sum_{k=1}^{N}(x_k)\sum_{k=1}^{N}(y_k)\right.\\
&\left.-N\sum_{k=1}^{N}(x_k y_k)\right)\Bigg)\\
&+\sum_{k=1}^{N}\left(p_k^{-1}\right)\left(\sum_{k=1}^{N}(x_k y_k)\sum_{k=1}^{N}(y_k)\right.\\
&\left.-\sum_{k=1}^{N}(x_k)\sum_{k=1}^{N}\left(y_k^2\right)\right)\Bigg)\\
\tilde{K}\tilde{\beta} &= \sum_{k=1}^{N}\left(x_k p_k^{-1}\right)\\
&\times\left(\sum_{k=1}^{N}(x_k)\sum_{k=1}^{N}(y_k)-N\sum_{k=1}^{N}(x_k y_k)\right)\Bigg)\\
&+\sum_{k=1}^{N}\left(y_k p_k^{-1}\right)\left(\sum_{k=1}^{N}\left(x_k^2\right)-\left(\sum_{k=1}^{N}(x_k)\right)^2\right)\\
&+\sum_{k=1}^{N}\left(p_k^{-1}\right)\left(\sum_{k=1}^{N}(x_k)\sum_{k=1}^{N}(x_k y_k)\right.\\
&\left.-\sum_{k=1}^{N}\left(x_k^2\right)\sum_{k=1}^{N}(y_k)\right)\\
\tilde{K}\tilde{\gamma} &= \sum_{k=1}^{N}\left(x_k p_k^{-1}\right)\left(\sum_{k=1}^{N}(x_k y_k)\sum_{k=1}^{N}(y_k)\right.\\
&\left.-\sum_{k=1}^{N}(x_k)\sum_{k=1}^{N}\left(y_k^2\right)\right)\\
&+\sum_{k=1}^{N}\left(y_k p_k^{-1}\right)\left(\sum_{k=1}^{N}(x_k)\sum_{k=1}^{N}(x_k y_k)\right.\\
&\left.-\sum_{k=1}^{N}\left(x_k^2\right)\sum_{k=1}^{N}(y_k)\right)+\sum_{k=1}^{N}\left(p_k^{-1}\right)\\
&\times\left(\sum_{k=1}^{N}\left(x_k^2\right)\sum_{k=1}^{N}\left(y_k^2\right)-\left(\sum_{k=1}^{N}(x_k y_k)\right)^2\right)
\end{aligned}
\tag{14}
$$

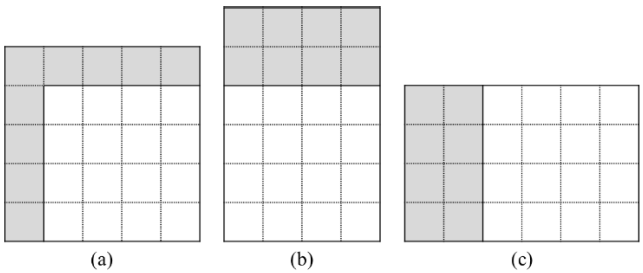


FIGURE 3. Reference pixel selection for PMCP: (a) as pixels surrounding current block, (b) as pixels in top box, and (c) as pixels in left box.

where N is the number of the reference pixels, (x_k, y_k) and p_k are image coordinates and a depth pixel value of kth reference pixel ($1 \le k \le N$), and

$$
\begin{aligned}
\tilde{K} &= N\sum_{k=1}^{N}\left(x_k^2\right)\sum_{k=1}^{N}\left(y_k^2\right)\\
&+2\sum_{k=1}^{N}(x_k y_k)\sum_{k=1}^{N}\left(x_k^2\right)\sum_{k=1}^{N}(y_k)\\
&-\sum_{k=1}^{N}\left(x_k^2\right)\left(\sum_{k=1}^{N}(y_k)\right)^2\\
&-\left(\sum_{k=1}^{N}(x_k)\right)\sum_{k=1}^{N}\left(y_k^2\right)\\
&-N\left(\sum_{k=1}^{N}(x_k y_k)\right)^2
\end{aligned}
\tag{15}
$$

The plane modeling by PMCP is performed by substituting $\tilde{\alpha}$, $\tilde{\beta}$, and $\tilde{\gamma}$ into (11) as follows:

$$
\tilde{P}_k = \left(\tilde{\alpha}x_k + \tilde{\beta}y_k + \tilde{\gamma}\right)^{-1}. \tag{16}
$$

The differences of PMCs between the block and the reference pixels are measured for 4×4 and 16×16 blocks. It can be expected that the accuracy of PMCP is higher as

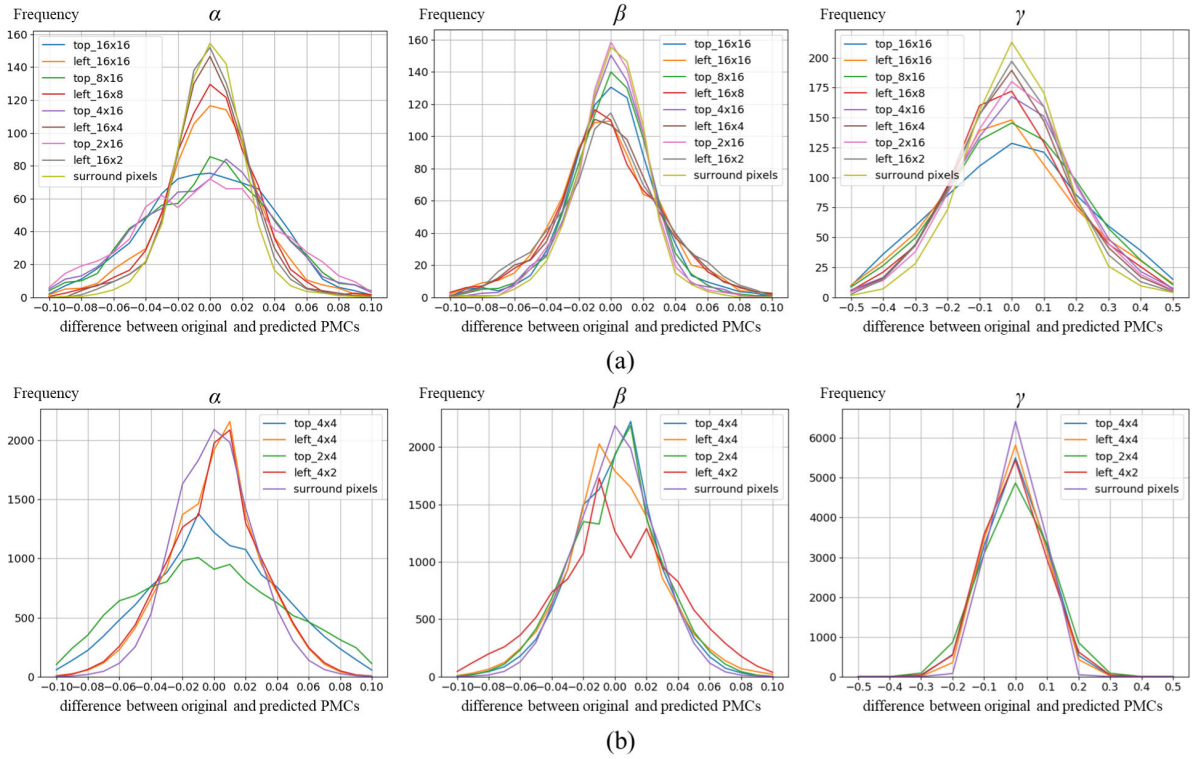


FIGURE 4. Distributions of differences between original and predicted PMCs: (a) for 16 × 16 block and (b) for 4 × 4 block.

the PMC differences are smaller. Fig. 4 and Table 2-3 show the distributions and the variances for the PMC differences, respectively. The variance of the PMC difference is the smallest when the depth pixels surrounding the block are selected as the reference pixels. Therefore, PMCP is most accurate in case of selecting the depth pixels surrounding the block.

The prediction errors of the depth pixels are measured for 4 × 4 and 16 × 16 blocks through the plane modeling by PMCP. Table 4 shows the comparison of the prediction errors depending on selection methods of reference pixels. For whole block sizes, the prediction accuracy is the highest when the depth pixels surrounding the block are selected as the reference pixels. The accuracy of the prediction is the next highest when the depth pixels of the $m \times 4$ top or the $4 \times n$ left boxes are selected. These results mean that the depth pixels surrounding the block should be thought as a top priority as the reference pixels for PMCP.

Fig. 5 (a) shows in case of including boundary between objects in the reference pixels when the depth pixels surrounding the block are selected. PMCP is inaccurate because some reference pixels are on a different surface. On the other hand, PMCP in Fig. 5 (b) is accurate because whole reference pixels and the depth pixel of the block are on the same plane. The accuracy of the plane modeling for the reference pixels through the predicted PMCs is high if whole reference pixels are on the same plane. The accuracy of PMCP is calculated as the mean squared error of the plane modeling for the reference pixels as follows:

$$MSE = \frac{1}{N}\sum_{k=1}^{N}\left(\left(\tilde{\alpha}x_k + \tilde{\beta}y_k + \tilde{\gamma}\right)^{-1} - p_k\right)^2. \quad (17)$$

TABLE 2. Variances of prediction errors (16 × 16).

Selection method for PMCP	$\sigma^2(\alpha)$	$\sigma^2(\beta)$	$\sigma^2(\gamma)$
surrounding pixels	0.0004	0.0003	0.0208
16×16 top block	0.0017	0.0005	0.0470
16×8 top block	0.0017	0.0005	0.0470
16×4 top block	0.0017	0.0004	0.0363
16×2 top block	0.0018	0.0005	0.0489
16×16 left block	0.0007	0.0011	0.0437
8×16 left block	0.0007	0.0011	0.0437
4×16 left block	0.0005	0.0010	0.0339
2×16 left block	0.0007	0.0012	0.0413

TABLE 3. Variances of prediction errors (4 × 4).

Selection method for PMCP	$\sigma^2(\alpha)$	$\sigma^2(\beta)$	$\sigma^2(\gamma)$
surrounding pixels	0.0005	0.0005	0.0018
4×4 top block	0.0020	0.0006	0.0056
4×2 top block	0.0035	0.0007	0.0083
4×4 left block	0.0007	0.0008	0.0039
2×4 left block	0.0007	0.0016	0.0054

TABLE 4. MSEs for plane modeling depending on selection methods.

16×16 block size		4×4 block size	
Selection method for PMCP	MSE	Selection method for PMCP	MSE
surrounding pixels	2.055	surrounding pixels	0.302
16×16 top block	6.293	4×4 top block	0.565
16×8 top block	5.432	4×2 top block	0.898
16×4 top block	4.736	4×4 left block	0.370
16×2 top block	5.772	2×4 left block	0.565
16×16 left block	4.538		
8×16 left block	3.623		
4×16 left block	3.058		
2×16 left block	3.328		

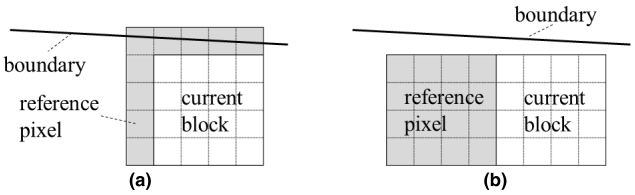


FIGURE 5. Selection of reference pixels in case of boundary between objects at top of block: (a) selecting depth pixels surrounding block and (b) selecting depth pixels in left box.

If the MSE in (17) is lower than a threshold T, then the reference pixels are regarded to be on the same plane. In this case, the corresponding PMCs are applied to the plane modeling through (16). Otherwise, other reference pixels are selected for PMCP. Table 4 shows that the accuracy of PMCP is high in the order of selecting the reference pixels from pixels surrounding the block, pixels in the $4 \times n$ left box, and pixels in the $m \times 4$ top box. If the MSEs of all of the sets of the reference pixels are larger than T, then the plane modeling is not applied to the corresponding block. The process of PMCP is as follows.

1. Calculating PMCs by PMCP for depth pixels surrounding block.
2. Calculating MSE by (17). If MSE $< T$, PMCs are set as 1.
3. Otherwise, calculating PMCs for depth pixels in left $4 \times n$ box.
4. Calculating MSE by (17). If MSE $< T$, PMCs are set as 3.
5. Otherwise, calculating PMCs for depth pixels in $m \times 4$ top box.
6. Calculating MSE by (17). If MSE $< T$, PMCs are set as 5.
7. Otherwise, plane modeling mode is not applied for corresponding block.
8. Predicting depth pixels by (16).

For applying the proposed method into color video coding standards, the intra mode by plane modeling with PMCP is added into the intra modes of color video coding standard. In this result, the number of the intra modes is increased. Other elements of bitstream syntax in video coding are not needed to be changed.

IV. SIMULATION RESULTS

We measure the improvement of the depth data coding through the proposed method comparing with conventional VVC. The proposed method can be also applied to other video coding standards. VVC Test Model (VTM) [22], which is a reference coding software for VVC, is modified for the proposed method so the plane modeling through PMCP is added to the intra modes of VVC. The intra mode is selected by the rate-distortion optimization in VTM. CABAC [21] is applied for symbol encoding. The depth data are encoded through QP in the range of 0∼51 which is full range of VTM. Whole frames are set as I frame so the frames are encoded by intra-picture prediction. T for PMCP is set to 500.

Since depth data represents the points at 3D space, the encoding distortion is evaluated to the differences of the position at 3D space, instead of pixel differences. In the previous studies [23]–[25] of depth data compression, the error of a reconstructed depth data is evaluated through RMSE. RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(e_{ij}\right)^2}, \tag{18}$$

where e_{ij} is an euclidian distance between the points at 3D world of original and encoded pictures. e_{ij} is calculated as follows:

$$\begin{aligned} e_{ij} &= \left(\bar{X}_{ij} - \tilde{X}_{ij}\right)^2 + \left(\bar{Y}_{ij} - \tilde{Y}_{ij}\right)^2 + \left(\bar{Z}_{ij} - \tilde{Z}_{ij}\right)^2 \\ &= \left[\frac{i}{f}\left(\bar{p}_{ij} - \tilde{p}_{ij}\right)\right]^2 + \left[\frac{j}{f}\left(\bar{p}_{ij} - \tilde{p}_{ij}\right)\right]^2 + \left(\bar{p}_{ij} - \tilde{p}_{ij}\right)^2 \\ &= f^{-2}\left(\bar{p}_{ij} - \tilde{p}_{ij}\right)^2\left(i^2 + j^2 + f^2\right) \end{aligned} \tag{19}$$

where $\left(\bar{X}_{ij}, \bar{Y}_{ij}, \bar{Z}_{ij}\right)$ and $\left(\tilde{X}_{ij}, \tilde{Y}_{ij}, \tilde{Z}_{ij}\right)$ are the 3D coordinates and $\bar{p}_{ij}$ and $\tilde{p}_{ij}$ are the depth pixel values of the pixels in the original and the encoded pictures, respectively.

In order to evaluate the best improvement of the proposed method in an ideal situation, the proposed method is compared with the conventional VVC for depth data which captures the plane surface.

Fig. 6 shows the comparison of the rate-distortion curves. Unlike to the result of the depth data encoding including PMC shown in Table 2, PMCP actually improves the encoding of the depth data.

Fig. 7 shows the selection ratios of the plane modeling depending on QP. The selection ratio is up to 53.68%.

Fig. 8 shows the prediction accuracy of intra-picture prediction by the plane modeling for each QP. The prediction accuracy increases more as QP is lower because the reference pixels for PMCP are less distorted by the quantization. Therefore, the improvement of the depth data coding performance is better through the proposed method as the QP is lower.

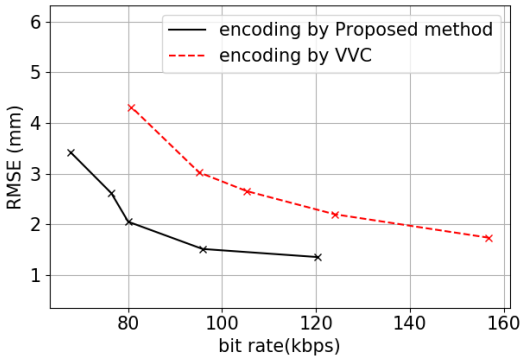


FIGURE 6. Comparison of rate-distortion curves for depth data capturing plane surface between proposed method and VVC.

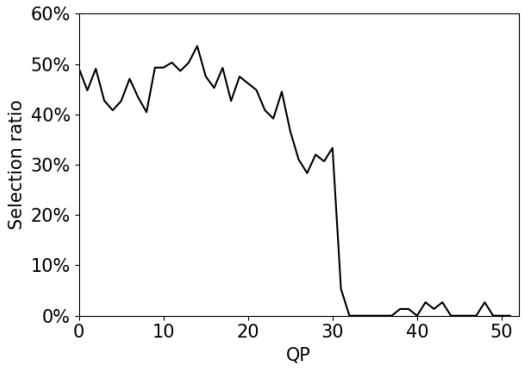


FIGURE 7. Selection ratio of plane modeling for depth data capturing plane surface.

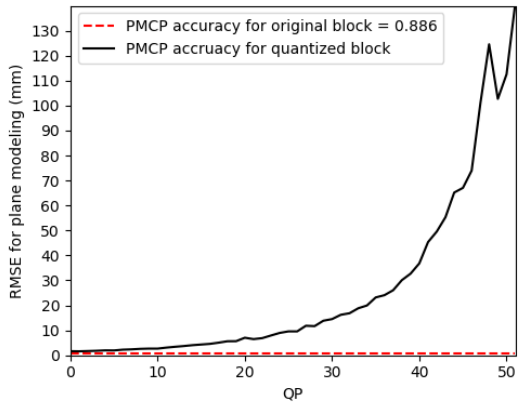


FIGURE 8. Intra-picture prediction error depending on QP.

Table 5-6 show the improvement rates of bit rates in same distortion and of distortions in same bit rate. In the ideal situation, the bit rates and RMSE are saving up to 56.86% and 27.83%, respectively. This result shows that the plane modeling by PMCP improves the depth data encoding in the plane surface area.

Fig. 9 shows depth data set [26]–[28] for the simulation. The depth data of 'table', 'round table', 'kitchen', and 'desk' are in [26], 'fire' and 'pumpkin' are in [27], and 'computer' and 'hat' are in [28]. The depth data for the simulation are captured by Microsoft Kinect. The focal length f of the depth camera is 526.370 mm. The resolutions and the frame rate of the depth data are 640 × 480 and 30 frame/s, respectively. First 60 frames of each depth data are encoded.

The execution times of the depth data encoding are measured to analyze the increase of computational complexity when the proposed intra mode is added into VVC.

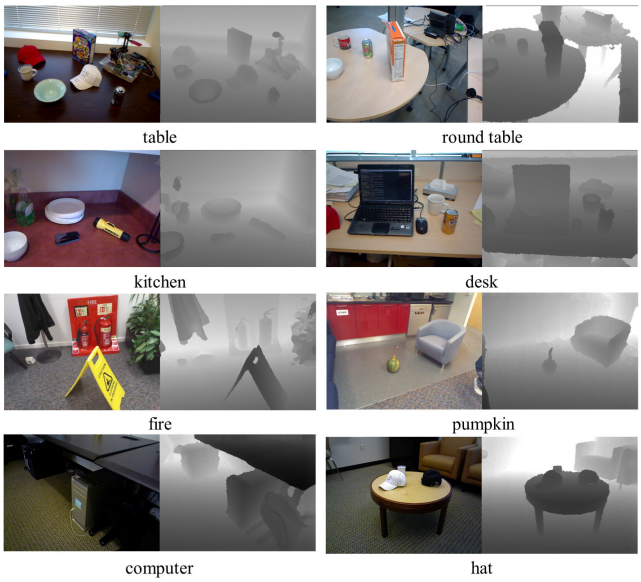


FIGURE 9. Depth videos for simulation.

TABLE 5. Distortion improvement for depth data of plane surface by proposed method.

Bit rate	50 kBPS	100 kBPS
Improvement rate for RMSE	56.86%	48.38%

TABLE 6. Bit rate improvement for depth data of plane surface by proposed method.

RMSE	3 mm	5 mm
Improvement rate for bit rate	27.83%	23.66%

TABLE 7. Increases in depth data coding time by adding proposed intra mode into VVC.

Source depth data	Increase rate of depth data coding time
table	13.10%
round table	18.10%
kitchen	8.60%
desk	21.30%
fire	21.20%
pumpkin	17.10%
computer	19.50%
hat	15.40%

TABLE 8. Distortion improvement for depth data by proposed method.

Source depth data	Bit rate	
	500 kBPS	1000 kBPS
table	9.01%	9.16%
round table	1.94%	2.48%
kitchen	12.07%	12.70%
desk	2.01%	2.29%
fire	2.04%	2.01%
pumpkin	2.84%	2.30%
computer	4.16%	3.15%
hat	2.34%	2.16%

TABLE 9. Bit rate improvement for depth data by proposed method.

Source depth data	RMSE	
	10 mm	15 mm
table	4.11%	3.17%
round table	4.40%	2.97%
kitchen	3.66%	2.38%
desk	3.35%	2.51%
fire	5.27%	4.36%
pumpkin	5.83%	5.65%
computer	6.76%	6.66%
hat	4.06%	3.76%

Table 7 shows the increase rate of depth data coding time. The average increase rate of depth coding time is about 16.8%.

Fig. 10 shows rate-distortion curves by the proposed method, VVC, H.265/HEVC, and Nenci's work [18] for the depth data coding. The proposed method improves the depth data coding comparing with the other methods for all of simulation depth data. In 'table' and 'kitchen', the bit rates are particularly improved for all of the RMSE.

Table 8-9 show the improvement rates of the bit rates and of the distortions for the simulation depth data. When the bit rates are fixed to 500 kbps and 1000 kbps, RMSEs are improved up to 12.07% and 12.70%, respectively. When RMSEs are fixed to 10 mm and 15 mm, the bit rates are improved up to 6.76% and 6.66%, respectively.

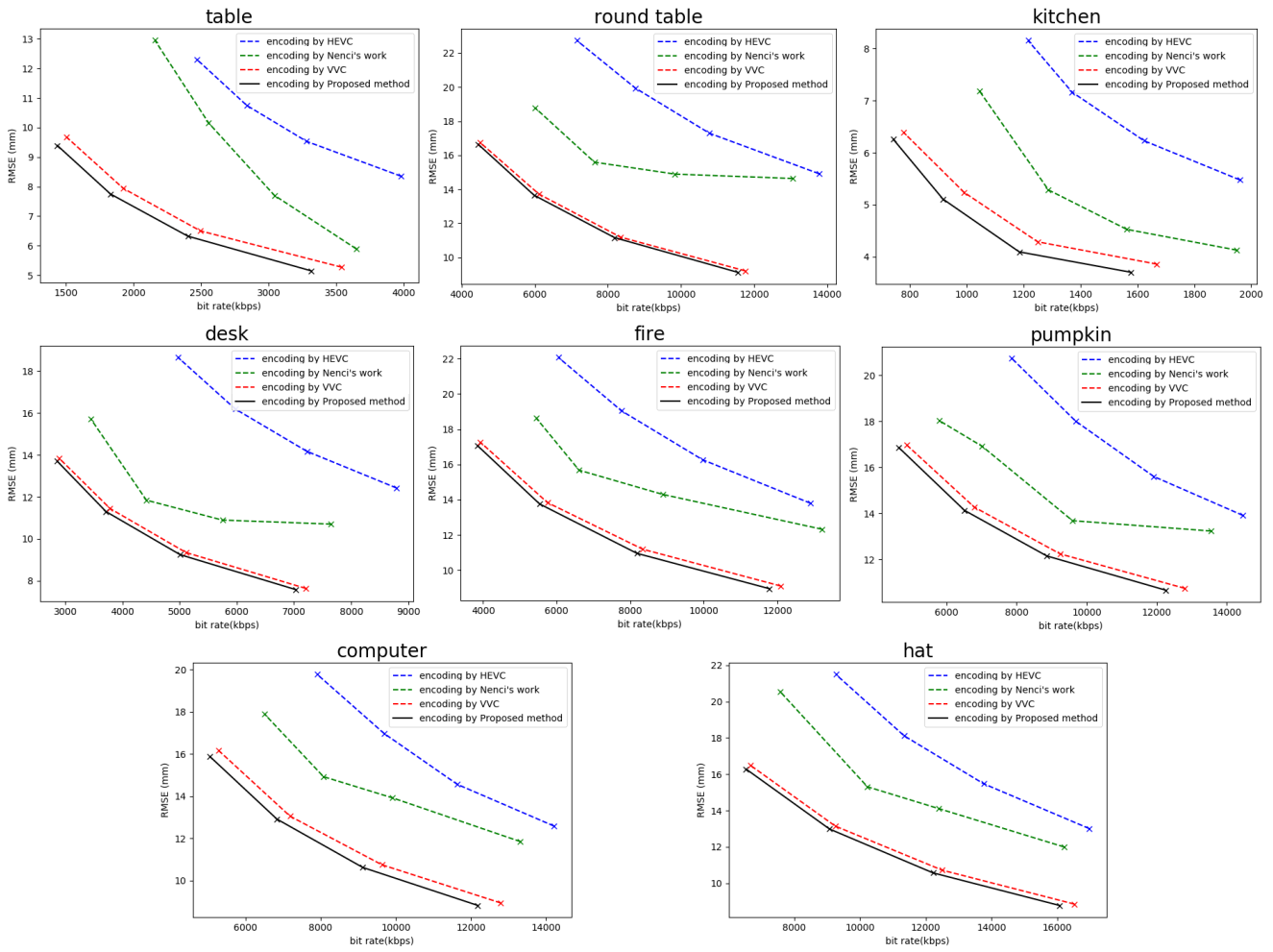


FIGURE 10. Comparison of rate-distortion curves for simulation depth data between proposed method and VVC.

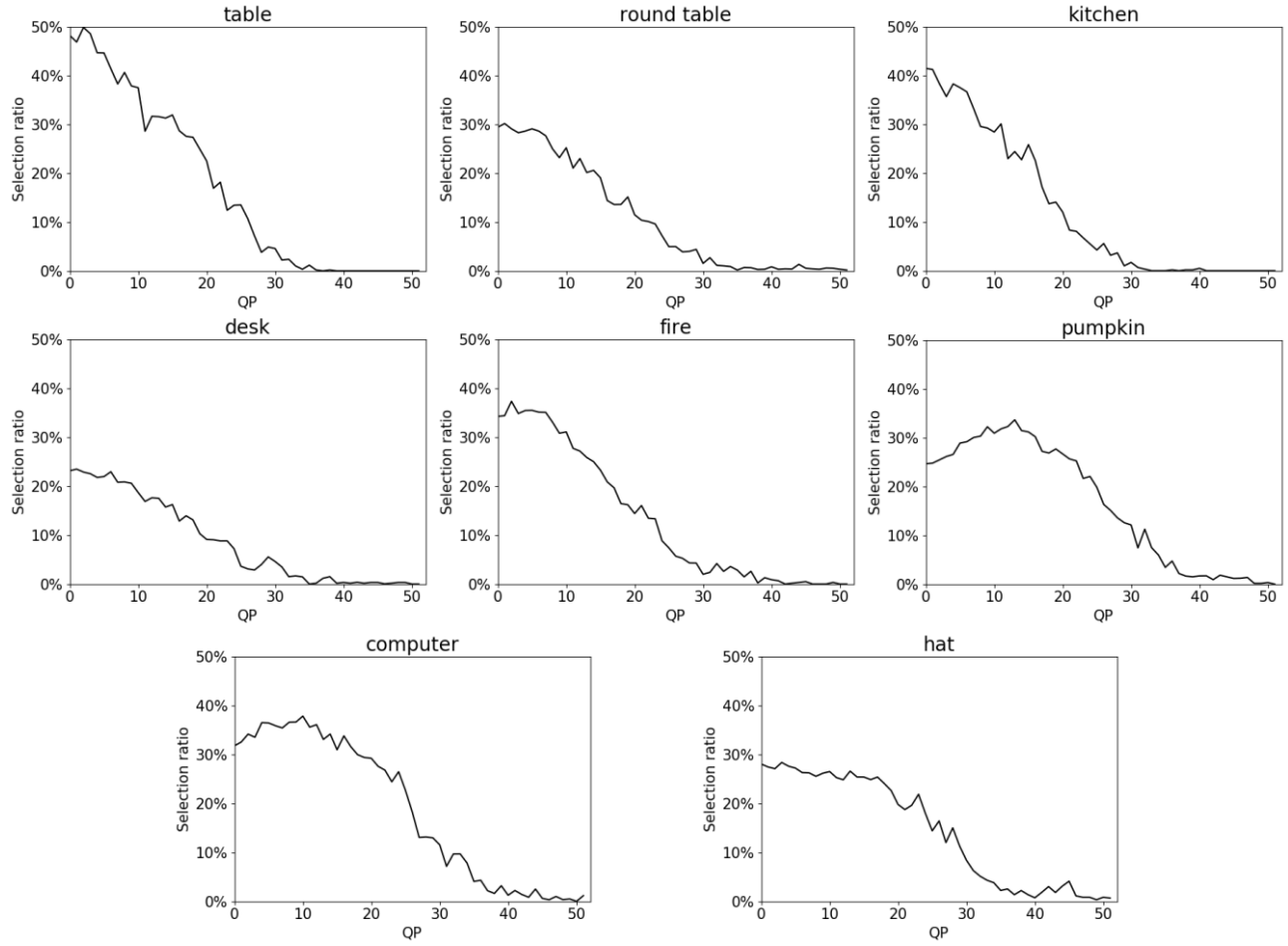


FIGURE 11. Selection ratio of plane modeling.

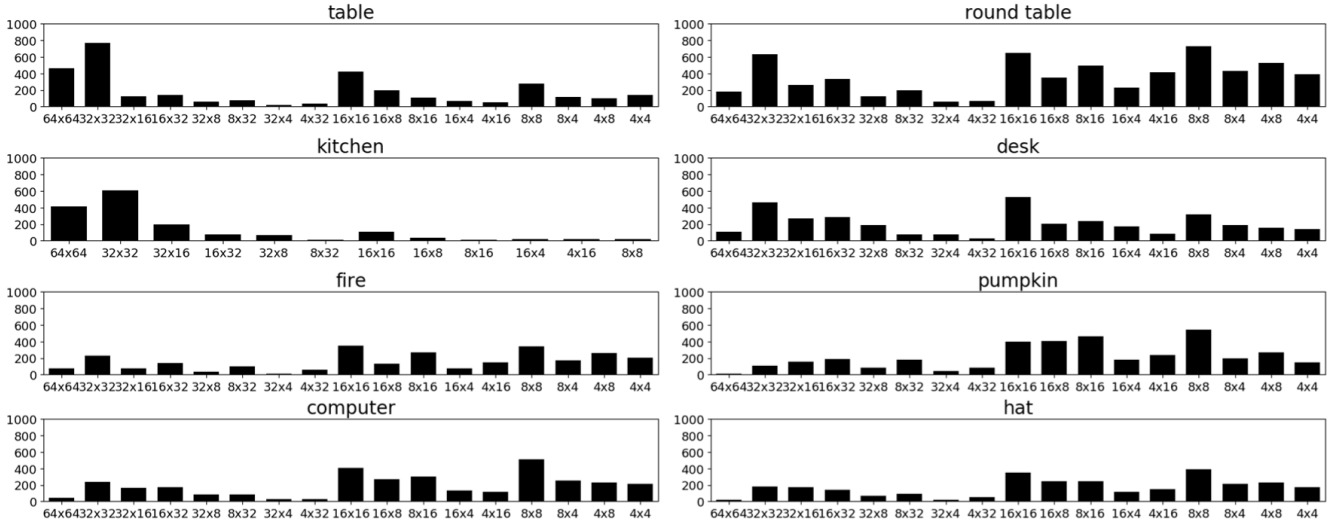


FIGURE 12. Frequency of the intra-predicted block size by the proposed method.

Fig. 11 shows the selection ratios of the plane modeling for each QP. The selection ratio of the plane modeling is higher as the QP is lower. In high QP, the reference pixels of PMCP have many distortions by the quantization, so the intra mode of the plane modeling less accurate than other intra modes.

Fig. 12 shows the frequencies of the intra-predicted block size by the proposed method. The frequencies of blocks whose width or height is more than 32 are large in the simulation videos 'table', 'kitchen', and 'desk', in which the number of objects is small and the background is simple.

V. CONCLUSION

In this article, we proposed the depth data coding method by the plane modeling through PMCP. PMCs of the block were predicted from the neighboring depth pixels. The reference pixels for PMCP were selected in the order of the depth pixels surrounding the block, the depth pixels in the left box, and the depth pixels in the top box. The accuracy of PMCP is calculated through the error of the plane modeling. If the plane modeling error was less than the certain threshold, the depth pixels were predicted by the plane modeling through the corresponding PMCs. In same bit rate and RMSE conditions, the RMSE and the bit rate were improved up to 12.70% and 6.76%, respectively, by the proposed method. The proposed method can apply to depth data captured by various type of depth camera and lidar. The proposed method can be extended by other surface modeling methods such as spherical modeling and ellipsoidal modeling.

REFERENCES

[1] Y. Zhao, M. Carraro, M. Munaro, and E. Menegatti, "Robust multiple object tracking in RGB-D camera networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 6625–6632.

[2] C. Y. Ren, V. A. Prisacariu, O. Kähler, I. D. Reid, and D. W. Murray, "Real-time tracking of single and multiple objects from depth-colour imagery using 3D signed distance functions," *Int. J. Comput. Vis.*, vol. 124, no. 1, pp. 80–95, Jan. 2017.

[3] M.-X. Jiang, X.-X. Luo, T. Hai, H.-Y. Wang, S. Yang, and A. N. Abdalla, "Visual object tracking in RGB-D data via genetic feature learning," *Complexity*, vol. 2019, pp. 1–8, May 2019.

[4] Y. Li, "Hand gesture recognition using Kinect," in *Proc. IEEE Int. Conf. Comput. Sci. Autom. Eng.*, Jun. 2012, pp. 196–199.

[5] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using Kinect sensor," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1110–1120, Aug. 2013.

[6] Y. Li, Q. Miao, K. Tian, Y. Fan, X. Xu, R. Li, and J. Song, "Large-scale gesture recognition with a fusion of RGB-D data based on the C3D model," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 25–30.

[7] Y. Sun, M. Liu, and M. Q.-H. Meng, "Improving RGB-D SLAM in dynamic environments: A motion removal approach," *Robot. Auto. Syst.*, vol. 89, pp. 110–122, Mar. 2017.

[8] W. G. Aguilar, G. A. Rodríguez, L. Álvarez, S. Sandoval, F. Quisaguano, and A. Limaico, "Visual SLAM with a RGB-D camera on a quadrotor UAV using on-board processing," in *Proc. Int. Work-Conf. Artif. Neural Netw.*, Jun. 2017, pp. 596–606.

[9] S.-K. Kwon, "Face recognition using depth and infrared pictures," *Nonlinear Theory Appl. IEICE*, vol. 10, no. 1, pp. 2–15, Jan. 2019.

[10] M. Domanski, O. Stankiewicz, K. Wegner, and T. Grajek, "Immersive visual media—MPEG-I: 360 video, virtual navigation and beyond," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, May 2017, pp. 1–9.

[11] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[12] *Versatile Video Coding*, document JVET-J1001, 2018.

[13] J. Fu, D. Miao, W. Yu, S. Wang, Y. Lu, and S. Li, "Kinect-like depth data compression," *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1340–1352, Oct. 2013.

[14] X. Wang, Y. A. Sekercioglu, T. Drummond, E. Natalizio, I. Fantoni, and V. Fremont, "Fast depth video compression for mobile RGB-D sensors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 4, pp. 673–686, Apr. 2016.

[15] X. Wang, Y. A. Şekercioğlu, T. Drummond, V. Frémont, E. Natalizio, and I. Fantoni, "Relative pose based redundancy removal: Collaborative RGB-D data transmission in mobile visual sensor networks," *Sensors*, vol. 18, no. 8, pp. 1–23, Jul. 2018.

[16] D. S. Lee and S. K. Kwon, "Intra prediction of depth picture with plane modeling," *Symmetry*, vol. 10, no. 12, pp. 1–16, Dec. 2018.

[17] S.-K. Kwon, A. Tamhankar, and K. R. Rao, "Overview of H.264/MPEG-4 part 10," *J. Vis. Commun. Image Represent.*, vol. 17, no. 2, pp. 186–216, Apr. 2006.

[18] F. Nenci, L. Spinello, and C. Stachniss, "Effective compression of range data streams for remote robot operations using H.264," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 3794–3799.

[19] O. Stankiewicz, K. Wegner, and M. Domanski, "Nonlinear depth representation for 3D video coding," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 1752–1756.

[20] R. Hartley and A. Zisserman, "Camera models," in *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge Univ. Press, 2000, pp. 153–177.

[21] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, Jul. 2003.

[22] *VVC Test Model*. Accessed: Apr. 29, 2020. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM.3

[23] M. Ruhnke, L. Bo, D. Fox, and W. Burgard, "Hierarchical sparse coded surface models," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 6238–6243.

[24] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5556–5565.

[25] X. Sun, H. Ma, Y. Sun, and M. Liu, "A novel point cloud compression algorithm based on clustering," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 2132–2139, Apr. 2019.

[26] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1817–1824.

[27] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 746–760.

[28] K. Lai, L. Bo, and D. Fox, "Unsupervised feature learning for 3D scene labeling," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 3050–3057.

DONG-SEOK LEE received the B.S. and M.S. degrees in computer software engineering from Dong-Eui University, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree with the Department of Computer Software Engineering. His research interests include image processing and video processing.

BYUNG-GYU KIM (Senior Member, IEEE) received the B.S. degree from Pusan National University, South Korea, in 1996, the M.S. degree from the Korea Advanced Institute of Science and Technology (KAIST), in 1998, and the Ph.D. degree from the Department of Electrical Engineering and Computer Science, KAIST, in 2004.

In March 2004, he joined the Real-Time Multimedia Research Team, Electronics and Telecommunications Research Institute (ETRI), South Korea, where he was a Senior Researcher. From February 2009 to February 2016, he was an Associate Professor with the Division of Computer Science and Engineering, Sun Moon University, South Korea. In March 2016, he joined the Department of Information Technology (IT) Engineering, Sookmyung Women's University, South Korea, where he is currently a Full Professor. He has published more than 250 international journal and conference papers, and patents in his field. His research interests include image and video signal processing for the content-based image coding, video coding techniques, 3-D video signal processing, deep/reinforcement learning algorithm, embedded multimedia systems, and intelligent information systems for image signal processing.

Dr. Kim is a Professional Member of ACM and IEICE. He also served or serves as an Organizing Committee Member for CSIP 2011, a Co-Organizer for CICCAT2016/2017, the 7th International Conference on Advanced Computing, Networking, and Informatics (ICACNI2019), and EAI 13th International conference on Wireless Internet Communications Conference (WiCON 2020), and a Program Committee Member for many international conferences. He has received the Special Merit Award for Outstanding Paper from the IEEE Consumer Electronics Society, at IEEE ICCE 2012, the Certification Appreciation Award from the SPIE Optical Engineering in 2013, and the Best Academic Award from the CIS in 2014. In ETRI, he developed so many real-time video signal processing algorithms and patents and received the Best Paper Award in 2007. He has been serving as a Professional Reviewer in many academic journals, including IEEE, ACM, Elsevier, Springer, Oxford, SPIE, IET, MDPI, IT&T, and so on. In 2007, he has served as an Editorial Board Member for the *International Journal of Soft Computing*, *Recent Patents on Signal Processing*, *Research Journal of Information Technology*, *Journal of Convergence Information Technology*, and *Journal of Engineering and Applied Sciences*. He has been serving as an Associate Editor for *Circuits, Systems and Signal Processing* (Springer), *The Journal of Supercomputing* (Springer), *Journal of Real-Time Image Processing* (Springer), *Heliyon Computer Science* (Cell Press), and *Applied Sciences* (MDPI). Since March 2018, he has been serving as the Editor-in-Chief for *Journal of Multimedia Information System* and an Associate Editor for IEEE Access journal. He has been serving as the Topic Editor for *Sensors and Electronics* (MDPI).

SOON-KAK KWON received the B.S. degree in electronic engineering from Kyungpook National University, in 1990, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), in 1992 and 1998, respectively. From 1998 to 2000, he was the Team Manager of the Technology Appraisal Center of Korea Technology Guarantee Fund. Since 2001, he has been a Faculty Member with Dong-Eui University, where he is currently a Professor with the Department of Computer Software Engineering. From 2003 to 2004, he was a Visiting Professor with the Department of Electrical Engineering, The University of Texas at Arlington. From 2010 to 2011, he was an international Visiting Research Associate with the School of Engineering and Advanced Technology, Massey University. His research interests include image processing, video processing, video transmission, depth data processing, and AI object recognition. He received the awards, such as the Leading Engineers of the World 2008 and the Foremost Engineers of the World 2008 from IBC and Best Papers from Korea Multimedia Society. His biographical profile has been included in the 2008–2014, 2017–2019 Editions of Marquis Who's Who in the World and the 2009/2010 Edition of IBC Outstanding 2000 Intellectuals of the 21st Century. He is also an Associate Editor of *NOLTA* journal (IEICE), a Topic Editor of *Electronics* journal (MDPI), and a Reviewer Board Member of *Signals* journal (MDPI). He is also working as a Reviewer of several journals, such as *Sensors*, *Applied Sciences*, *Information*, *Symmetry*, *Entropy*, IEEE Transactions on Circuits and Systems for Video Technology, and IEEE Access.