Depth Video Coding Using Adaptive Geometry Based Intra Prediction for 3-D Video Systems

Min-Koo Kang, *Student Member, IEEE*, and Yo-Sung Ho, *Senior Member, IEEE*

***Abstract*—Depth video coding is an essential part of 3-D video processing systems. Specifically, object boundary regions are important in depth video coding since these regions significantly affect the visual quality of a synthesized view. In this paper, we propose an efficient depth video coding method to determine precise intra prediction modes and thereby reduce the loss of boundary information. To achieve this objective, we analyze and exploit statistical and geometric characteristics of the depth video. Experimental results subsequently show that the proposed method performs better than the original intra prediction of H.264/AVC in terms of bit savings and rendering quality.**

***Index Terms*—Depth video coding, geometry-based block partitioning, intra prediction, 3-D video system.**

I. Introduction

NEXT-GENERATION broadcasting technologies such as 3-D video (3DV) and ultra high definition TV (UHDTV) have been widely researched as a combined system in attempts to expand the experiences of viewers far beyond the capabilities of current 2-D televisions. This system provides a more immersive sense of realism and the function of free viewpoint navigation [1], and as such various applications have been developed in conjunction with related fields such as computer graphics. One of the most popular applications is free viewpoint video (FVV) [2], which allows viewers to select an arbitrary viewpoint and direction within a certain range. Another application is 3-D television (3DTV) [3], which provides viewers with the 3-D depth of a captured scene by copying a binocular vision system.

One difficulty of these 3DV applications pertains to the vast amount of input data that must be compressed and transmitted to viewers. 3-D contents are usually generated by multi-camera setups or via 3-D modeling; thus, the amount of input data is significantly larger than that of 2-D contents. In this point of view, both efficient 3-D scene representation and 3-D content compression are essential in 3DV systems, with video plus depth (V+D) being one of the most common data formats [4] because of its suitability for 3-D scene rendering and data compression. Moreover, this format can be extended to multi-view scenarios for advanced applications such as auto-stereoscopic displays [4].

Nowadays, various approaches have been proposed in attempts to compress such data. One approach is to focus on reducing the bitrate for depth video compression. In this approach, common information is shared between color and depth videos to reduce data redundancy. As examples, Oh *et al.* [5] proposed a motion vector sharing method between color and depth videos, and Na *et al.* [6] more recently proposed a view synthesis prediction method for depth video coding. This approach appears adequate with respect to bit-saving, though it has strict constraints pertaining to encoding order and data structure, thereby limiting its applicability in 3DV systems.

Another approach is to focus on the rendering quality of synthesized views since the main function of depth data in 3DV systems is to assist in the synthesis of high quality virtual viewpoint views. To date, a number of region-adaptive coding methods have been introduced based on this approach. One example is to increase rendering quality via the special handling of the object boundary regions of depth video, since the boundary information significantly affects the rendering quality of synthesized views [7], [8]. This method results in good rendering quality, but the compression ratio is less than adequate. To increase the compression power, regions whose quality does not significantly affect rendering quality can be encoded at a lower bitrate [9], though this too cannot be a fundamental solution since it cannot prevent quality degradation of the object boundary regions.

Therefore, a technique that preserves boundary information well and gives a sufficient compression gain is strongly required for 3DV systems. Previously, the geometry-adaptive block partitioning method by Dai *et al.* [10] was applied to color video coding, and Kang *et al.* [11] proposed a modified version of it for depth video coding. However, this early version required further improvements since it was too dependent on the geometric structures of the input sequences. In order to resolve this problem, an improved version of the algorithm was proposed [12]. This version is geometrically more robust since it adaptively selects a partitioning function according to the geometric structure of the neighboring blocks.

In this paper, we propose a depth video coding method based on this latest version of the adaptive geometry-based intra prediction method. The proposed method is optimized and evaluated via a series of test sequences and publicly provided encoders, which includes a computational complexity analysis. In addition, notable characteristics of the depth video are observed and analyzed, and strategies for efficient depth video coding are subsequently suggested. In order to evaluate the proposed method, bitrates of the compressed depth videos, qualities of the depth videos and synthesized virtual views, and subjective qualities of the virtual views are also compared.

The remainder of this paper is organized as follows. In Section II, we will review related works in order to better explain our proposed method. The proposed method is then

Manuscript received October 31, 2010; revised April 08, 2011 and July 11, 2011; accepted September 12, 2011. Date of publication September 22, 2011; date of current version January 18, 2012. This work was supported in part by Samsung Electronics Co., Ltd. and in part by MKE under the ITRC support program supervised by NIPA (NIPA-2011-(C1090-1111-0003)). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ming-Ting Sun.

The authors are with the School of Information and Communications, Gwangju Institute of Science and Technology (GIST), 500-712 Gwangju, Korea (e-mail: minkoo@gist.ac.kr; hoyo@gist.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2011.2169238

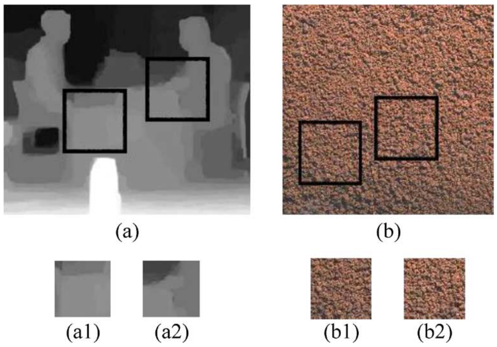


Fig. 1. Unique level distribution of depth video: (a) depth image and (b) texture image. (a1), (a2) Content in depth. (b1), (b2) Content in texture.

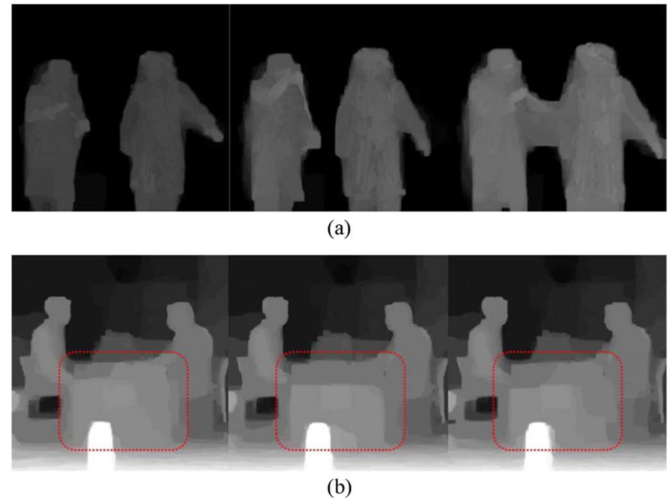


Fig. 2. Temporal inconsistency phenomenon on MPEG test sequences: (a) moving objects of "Pantomime" and (b) static objects of "Laptop".

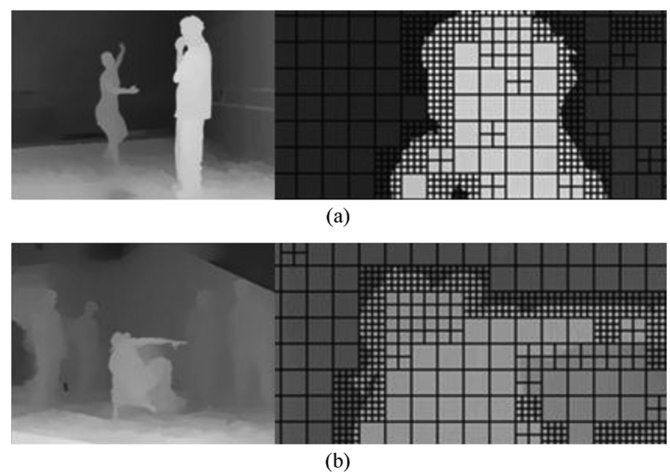


Fig. 3. Depth videos and their results of quadtree-based block partitioning: (a) "Ballet" sequence and (b) "Breakdancers" sequence.

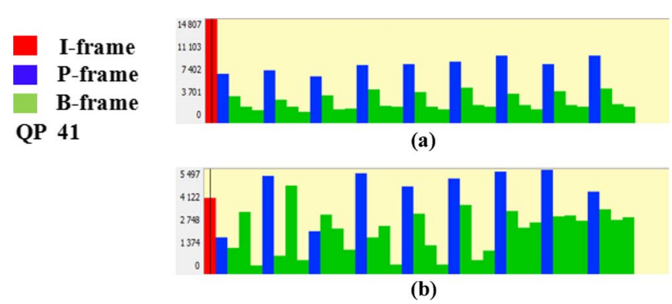


Fig. 4. Comparisons of bitrate per frame for "Pantomime" sequence: (a) results of color video coding and (b) results of depth video coding.

described in Section III, and in Section IV, we present coding performances and experimental results of the proposed method. Finally, we conclude this paper in Section V.

II. Related Works

In this section, we first explain unique characteristics of depth video that differ from those of color video. Next, we present two strategies for efficient depth video coding based on these observed differences and then introduce the concept of geometry-based block partitioning, which is a key component of our proposed method.

A. Characteristics of Depth Video Coding and Strategies

Most depth data has the same format as color video, YUV 8-bit, and thus can be compressed using color video codecs such as H.264/AVC. However, compared to color video, depth video has a number of different characteristics, which make color video codec-based techniques for depth video coding less efficient—in spite of the same format. An obvious difference is the depth-level distribution. Depth video has no texture since it consists solely of the distance information between the capturing camera and an object. Therefore, most regions within an object have very similar levels. Though this region is referred to as a "homogeneous region," regions around object boundaries show abrupt level changes (cf. Fig. 1).

Another difference of depth video is its lower temporal consistency than color video. In practice, depth data often show this phenomenon, except for depth data generated by computer graphics, since current depth capturing devices are not sufficiently precise, or the results of depth estimation methods such as stereo matching are not satisfactory (cf. Fig. 2).

Consequently, the unique depth-level distribution results in characterized quadtree-based block partitioning, such that most homogeneous regions are coded with the simplest and largest prediction modes (e.g., SKIP, Inter 16 × 16, or Intra 16 × 16), whereas boundary regions are coded with the smallest but simplest prediction modes (e.g., Intra 4 × 4); in Fig. 3, each rectangular represents the size of a prediction block.

In addition, low temporal consistency phenomena are directly connected to less efficient inter prediction (cf. Fig. 4). Usually, a B-frame requires a lower bitrate than an I-or P-frame in color video coding since it reduces temporal redundancy via its bi-directional inter prediciton scheme, though this is not the case here since depth video has a lower temporal consistency as shown in Fig. 2.

Taking account into these different characteristics of depth video coding, we now suggest strategies for efficient coding with respect to bit-saving and rendering quality. First, we need to exploit spatial correlations as much as possible, because exploiting temporal correlations in depth video is quite difficult. Second, we have to focus on the boundaries because a relatively

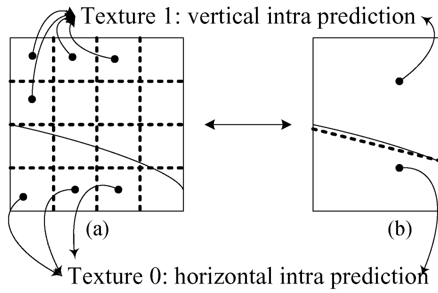


Fig. 5. Block partitioning comparison when a block contains a piece-wise-smooth 2-D signal: (a) quadtree-based method and (b) geometry-based method.

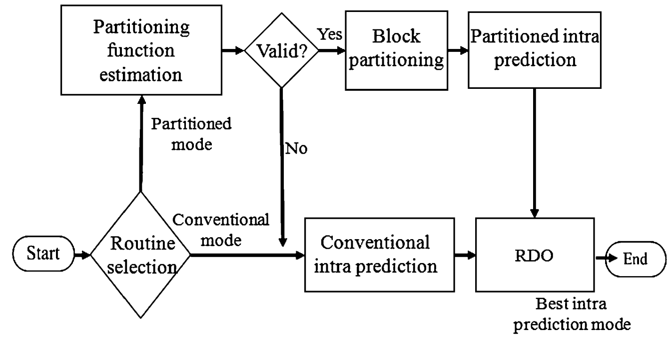


Fig. 6. Block diagram of the proposed adaptive geometry-based intra prediction.

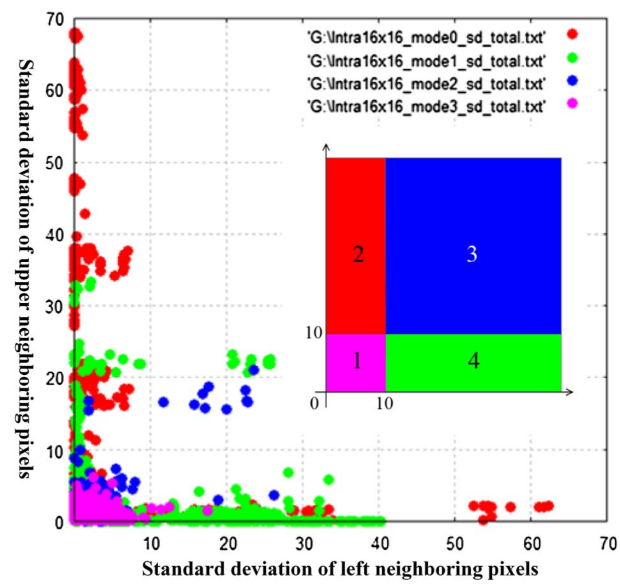


Fig. 7. Standard deviation distributions according to each best Intra 16 × 16 mode and its divisions to connect PIPE types.

large number of residual data and quantization errors occur around these regions.

B. Geometry-Adaptive Block Partitioning

In geometry-adaptive block partitioning [10], a prediction block for the current block is divided into three regions via an estimated line. Then, each region is independently predicted based on available neighboring pixels. After this, the predicted regions are combined into a single prediction mode instead of dividing the current block into several sub-blocks, as in conventional quadtree-based block partitioning (cf. Fig. 5). Eventually, the used information for the current block partitioning is encoded as side information.

This block partitioning method is generally efficient for depth video coding since it provides an efficient intra prediction mode around object boundaries when the current block can be locally modeled as a piecewise-smooth 2-D signal, and depth video often satisfies the ideal condition because of its simple level distribution. Moreover, it is possible to increase the rendering quality by increasing the quality of object boundaries of compressed depth video since the rendering quality significantly depends on the quality of object boundaries [13]. Therefore, we can improve both bit-saving and the rendering quality by applying the geometry-adaptive block partitioning scheme to depth video coding.

In this paper, we modified the previous geometry-adaptive block partitioning method and optimized it by exploiting unique characteristics of the depth video. As such, the main differences between the previous method and the proposed method are as follows. First, the proposed method does not require side information for block partitioning at the decoder side since it estimates partitioning information from the geometric structures of previously reconstructed and available neighboring blocks. Second, the proposed intra prediction modes, partitioned and independently predicted, adaptively alternate between the conventional intra prediction modes of H.264/AVC.

III. Proposed Algorithm

In this section, we further explain our proposed method. The analyzed characteristics of depth video coding enable us to adopt previous geometry-adaptive block partitioning to depth video coding. First, we can easily estimate a partition function from the geometric structures of available neighboring blocks, and thereby exploit the high spatial correlation of the depth video signal. Second, we can quickly determine whether or not the current block contains a boundary from the information of neighboring pixels by inspecting their continuities.

The block diagram of the proposed method is presented in Fig. 6. The proposed method starts by investigating the standard deviations of the left and upper neighboring pixels to determine whether or not the encoder uses the proposed method. Since depth video signals have a high spatial correlation, the most probable Intra 16 × 16 prediction mode and extrapolation directions can then be predicted from the standard deviations.

Fig. 7 shows the distributions of standard deviations according to each Intra 16 × 16 prediction mode when each mode is selected as the best prediction mode for the “Ballet” depth sequence, where the x-axis is the standard deviation of the left neighboring pixels, and the y-axis is the standard deviation of the upper neighboring pixels. From these results, we can confirm that each prediction mode has a biased distribution in a certain direction. Thus, we can predict the most probable Intra 16 × 16 prediction mode based on the positions of the standard deviations. For convenience, we divided the standard deviation plane into four divisions, where each division maps a partitioned intra prediction mode extension (PIPE) type and position of neighboring block to be referred for the partition function estimation.

The PIPE types define sets of Intra 16 × 16 prediction modes to be constructed by the proposed method, and Table I shows

TABLE I
PIPE TYPE AND NEIGHBORING BLOCK POSITION

Division	PIPE type	Neighbor Block
1	Vertical, Horizontal, DC, Plane	Left, Upper
2	Vertical + *Vertical'*, *Median'*, *Plane'*	Upper
3	*Vertical'*, *Horizontal'*, *Median'*, *Plane'*	Left, Upper
4	Horizontal + *Horizontal'*, *Median'*, *Plane'*	Left

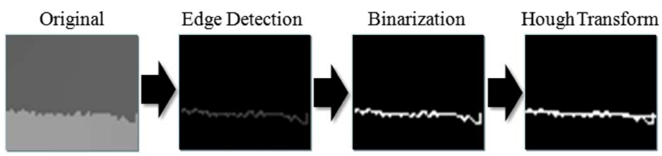


Fig. 8. Process-chain for line detection using the Hough transform.

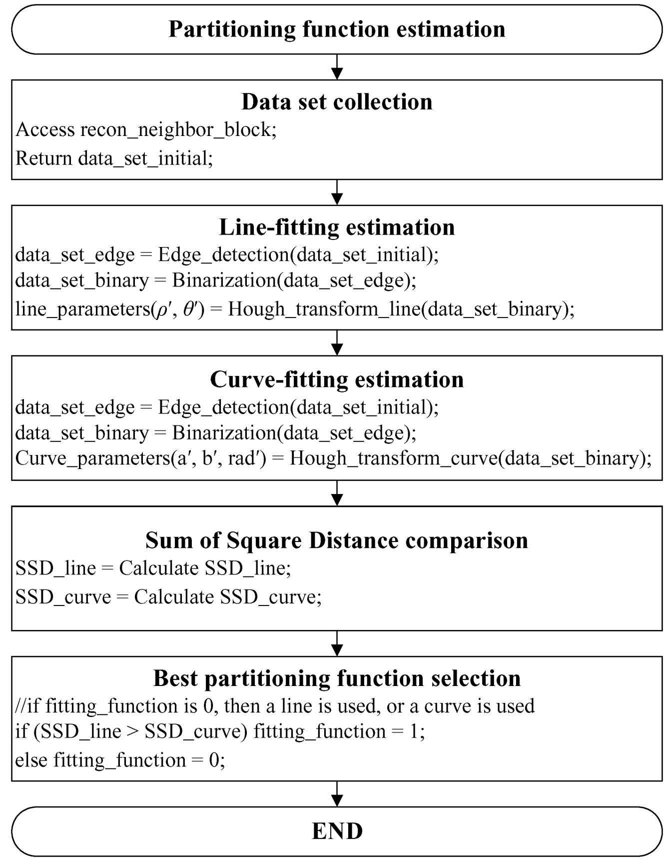


Fig. 9. Flowchart for partitioning function estimation process.

the PIPE types connected to each division, where the newly proposed intra 16 × 16 prediction modes are written in italics.

A. Adaptive Partitioning Function Estimation

A partitioning function is required for dividing the current block into two regions. After selecting the most probable prediction mode and position of the neighboring block, the partition functioning is estimated from the geometric structures of the selected neighboring block. Fig. 8 illustrates the general process for line detection using the Hough transform [14]. In additional, the proposed method performs a curve-detection/fitting procedure to obtain a more natural boundary. Thus, in the proposed method, the current block is divided by an estimated line or a curve function.

In Fig. 9, the flowchart describes the selection process for obtaining the best partitioning function. First, each process for an estimation of the line-fitting or curve-fitting function is simultaneously performed using geometric structures from the previously reconstructed neighboring blocks. Next, the sum of the square distance (SSD) of each fitting function is calculated. Between the two candidate functions, the function giving a smaller SSD is selected as the best partitioning function.

After the partitioning function estimation, the validity of the partitioning function is determined by investigating the position of the neighboring pixel that the partition function passes through. The partitioning function must pass one of the referred neighboring pixels to be valid because the current block should contain a boundary that passes through a neighboring pixel before we apply the proposed method. As such, if the partitioning function is invalid, the process is terminated.

B. Partitioning the Current Block

In this step, partitioning equations are established using the estimated partitioning function and its parameters. If a line function is estimated, two line parameters (ρ', θ') represent the line function, and partitioning equations are established using these two parameters. Similarly, if a curve function is estimated, three parameters $(\mathrm{a}', \mathrm{b}', \mathrm{rad}')$ represent the curve function, and partitioning equations are established using these three parameters for partitioning the current block [cf. (1)–(3)]. Equations (1) and (2) are partitioning equations for line-based partitioning when the left and upper neighboring blocks are referred, respectively. Similarly, (3) is the partitioning equation for the curve-based partitioning where the constants w and h represent the width and height of the referred neighboring block, respectively.

Using these partitioning equations, the position of the current pixel $(\mathrm{x_i}, \mathrm{y_j})$ is placed into one of two regions (negative or positive region) according to the output sign of the partitioning equation (cf. Fig. 10):

$$f(x_i, y_j) = (x_i + w)\sin\theta' + y_j\cos\theta' - \rho' \quad (1)$$

$$f(x_i, y_j) = x_i\sin\theta' + (y_j + h)\cos\theta' - \rho' \quad (2)$$

$$f(x_i, y_j) = (x_i + w - a')^2 + (y_j + h - b')^2 - +rad'. \quad (3)$$

C. Partitioned Intra Prediction Mode Extension

Once the current block is divided into two regions, each region is independently predicted from available neighboring pixels by various directional prediction methods. Then, the predicted regions are used to construct a single partitioned intra 16 × 16 prediction mode.

In this manner, a set of partitioned intra prediction modes is produced according to PIPE type, and the shape of the partitioned current block is determined using an estimated partition function. The partitioned intra prediction modes alternate between conventional Intra 16 × 16 prediction modes. Thus, no new bits-to-signal prediction modes are necessary. Fig. 11

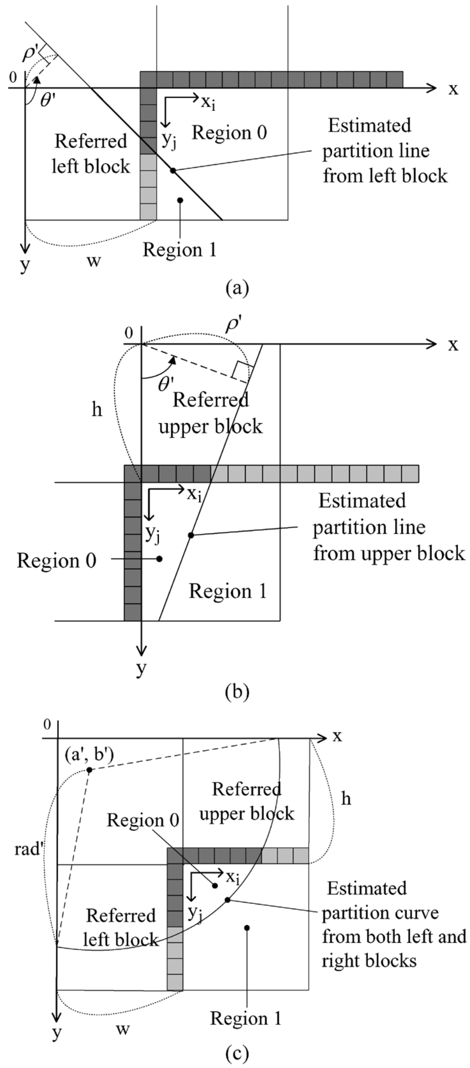


Fig. 10. Partitioning the current block into two regions: (a) horizontal line-based block partitioning, (b) vertical line-based block partitioning, and (c) curve-based block partitioning.

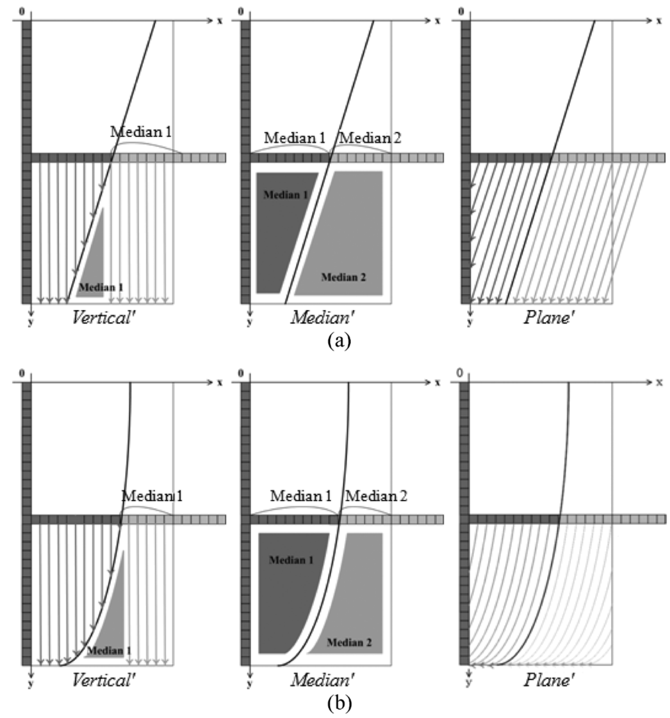


Fig. 11. Partitioned intra prediction modes when the PIPE is type 2: (a) line-based block partitioning and (b) curve-based block partitioning.

presents examples of the partitioned intra prediction modes when the PIPE is type 2.

In the case of PIPE type 2, three line-based or curve-based partitioned intra prediction modes $(Vertical', Median', Plane')$ are generated depending on the selected partitioning function, instead of using the conventional horizontal, DC, and plane intra prediction modes (cf. Fig. 11). First, the $Vertical'$ mode predicts each partitioned region in the vertical direction, and it alternates between the conventional horizontal prediction mode (mode number 1) where the region in which the vertical prediction cannot be obtained is predicted using the median value of its neighboring pixels. Similarly, the $Median'$ mode predicts each partitioned region using the median value of its available neighboring pixels, and it alternates between the conventional DC mode (mode number 2). Finally, the $Plane'$ mode predicts each partitioned region in the same direction of the partitioning function slope, and it alternates between the conventional plane prediction mode (mode number 3).

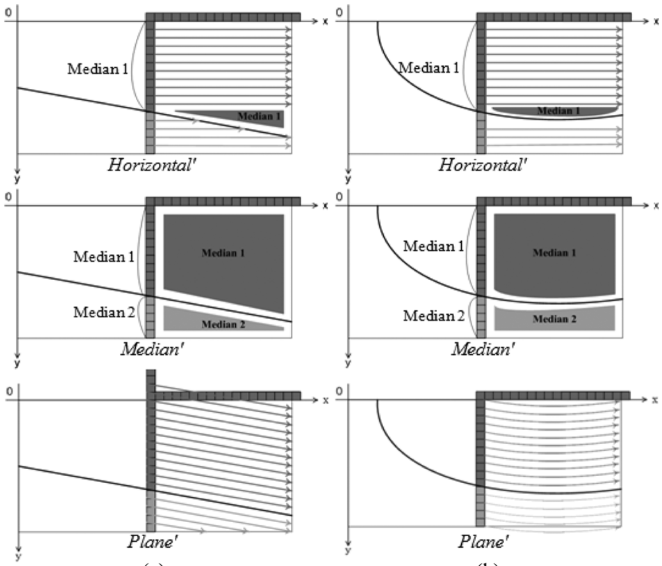
Note that these modes are rarely selected as the best Intra 16 $\times$ 16 mode when the standard deviations belong to division 2 from Fig. 7. Thus, we can allocate the mode numbers for the conventional modes to the partitioned intra prediction modes; thus, the most probable mode (mode number 0) plus the three partitioned intra prediction modes are used to construct the final candidates for the best Intra 16 $\times$ 16 prediction mode.

In the same manner, three partitioned intra prediction modes $(Horizontal', Median', Plane')$ are generated when the PIPE is type 4 (cf. Fig. 12). The new modes alternate between the conventional prediction modes (vertical, DC, and plane), except for the most probable prediction mode (horizontal). In the case of PIPE type 1, we cannot determine the most probable prediction mode because all prediction modes are randomly selected as the best Intra 16 $\times$ 16 prediction mode; thus, the conventional intra prediction is used as it is.

On the other hand, four partitioned intra prediction modes $(Vertical', Horizontal', Median', Plane')$ are generated when the PIPE is type 3 (cf. Fig. 13), and these alternate between the all conventional modes (vertical, horizontal, DC, and plane).

IV. Experimental Results and Analysis

In spite of the recent activity of the Moving Picture Experts Group (MPEG) 3-D video group in attempts to standardize 3DV systems [15]–[17], various depth data and image or video compression tools such as JPEG2000 are still being widely researched. Therefore, performance comparisons of related works are troublesome in this research field. In this paper, we first compared the comprehensive depth map coding performances of the JPEG2000 encoder (Kakadu software) [18], platelet-based encoder [19], and the proposed method using the first frames of "Ballet" and "Breakdancers", generated by the Interactive Visual Group at Microsoft Research [20]. Then, to compare the original H.264/AVC and proposed encoders, JM reference software version 14.2 was used [21] to show specific



(a) (b)

Fig. 12. Partitioned intra prediction modes when the PIPE is type 4: (a) line-based block partitioning and (b) curve-based block partitioning.

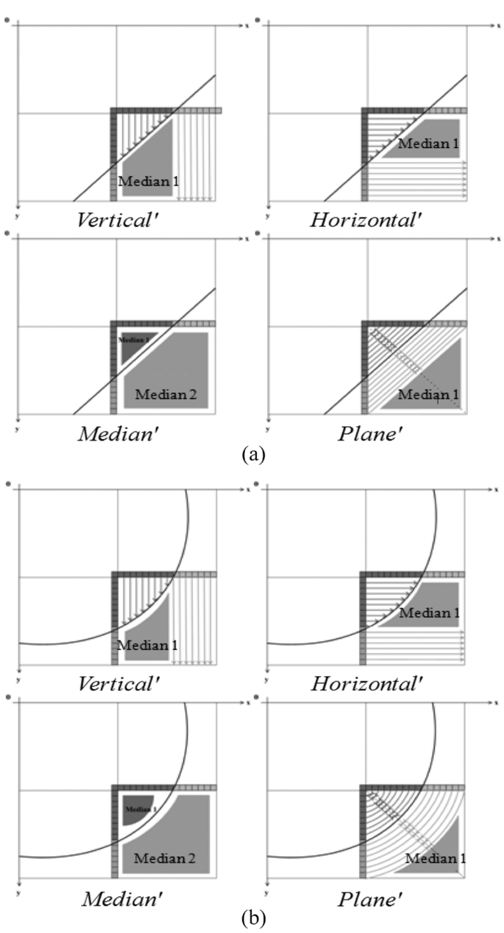


Fig. 13. Partitioned intra prediction modes when the PIPE is type 3: (a) line-based block partitioning and (b) curve-based block partitioning.

contributions of the proposed method, including the coding effects for quality of synthesized views.

Fig. 14 and Table II show the depth data, their properties, and simulation conditions used in the second experiment. The

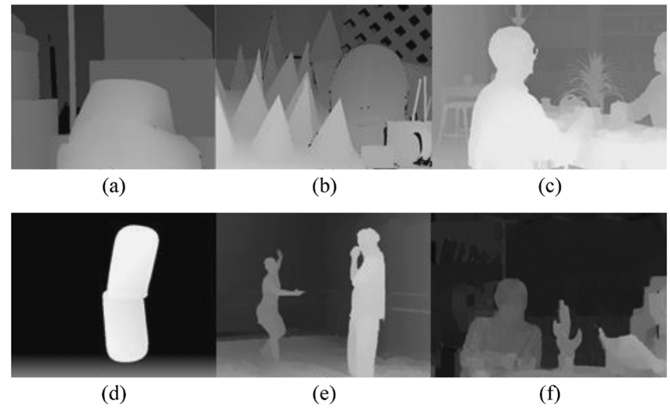


Fig. 14. Test depth sequences: (a) "Lampshade2", (b) "Cones", (c) "Beer_garden", (d) "Mobile", (e) "Ballet", and (f) "Newspaper".

TABLE II
SIMULATION CONDITIONS

	Provider	Name/ Viewpoint	Image Property
Test sequences	Middlebury	Lampshade2/ 5th	433×370, depth map
		Cones/ 2nd	1600×1006, depth map
	Philips	Beer_garden/ 5th	1920×1080, 30 fps, 150 frames
		Mobile/ 5ht, 7th	720×540, 30 fps, 200 frames
	MSR	Ballet/ 3rd, 5th	1024×768, 30 fps, 100 frames
	GIST	Newspaper/ 2nd, 4th	1024×768, 30 fps, 300 frames
GOP structure	Intra only frame		
QP	40, 42, 44, 46		
Common	FRExt profile, CABAC		

"Lampshade2" and "Cones" sequences are often used as the ground truth for evaluating stereo matching algorithms [22]. Thus, these depth maps are quite reliable in spite of their small holes. "Beer_garden" and "Mobile" sequences are computer graphics, and thus show the most satisfying depth quality. The depth sequences of "Ballet" and "Newspaper" [23] were generated by stereo matching algorithms, and hence contain relatively noticeable depth errors.

To evaluate the coding performances, we used terms of the Bjonteggard Delta BitRate (BDBR) and Bjonteggard Delta PSNR (BDPSNR) [24], in addition to the view synthesis reference software (VSRS2.0) [25] and MPEG's EE4 2-view test scenario [17] for "Mobile", "Ballet", and "Newspaper" in order to confirm the coding effects for the synthesized views. The synthesis results are calculated from a summation of bitrate pairs from the left and right depth videos and PSNR terms of the synthesized views.

Comprehensive coding performances of related works are compared in Fig. 15, which shows the rate-distortion (R-D) curves for the proposed encoder, platelet-based encoder, and JPEG2000 encoder for the "Ballet" and "Breakdancers" depth maps. Although the PSNR values do not exactly reflect the quality of the reconstructed images, we can see that the proposed method is the best among all encoders; the coding gain is more than 4 dB for "Ballet" and more than 2.5 dB for "Breakdancers".

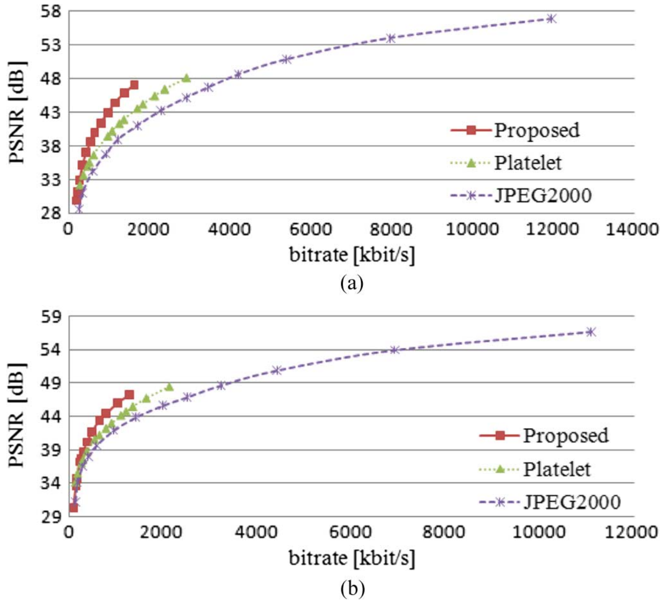


Fig. 15. R-D curves for depth map coding: (a) "Ballet" 5th viewpoint, frame 0, and (b) "Breakdancers" 5th viewpoint, frame 0.

TABLE III
EXPERIMENTAL RESULTS

Test sequence	View -point	BDBR [%]	BDPSNR [dB]	Encoder ΔTime [%]	Decoder ΔTime [%]
Lampshade2	5th	-7.960	0.563	+8.67	+19.81
Cones	2nd	-4.400	0.250	+6.34	+43.63
Beer_garden	5th	-2.990	0.248	+3.33	+16.88
Mobile	5th	-1.780	0.183	+4.49	+6.14
	7th	-1.310	0.179	+5.69	+7.96
	6^{th} (syn.)	-3.480	0.304		
Ballet	3rd	-1.600	0.116	+4.29	+11.72
	5th	-2.200	0.172	+4.50	+13.54
	4^{th} (syn.)	-4.570	0.224		
Newspaper	2nd	-3.300	0.266	+5.26	+21.49
	4th	-1.870	0.120	+5.44	+22.76
	3^{rd} (syn.)	-3.670	0.120		

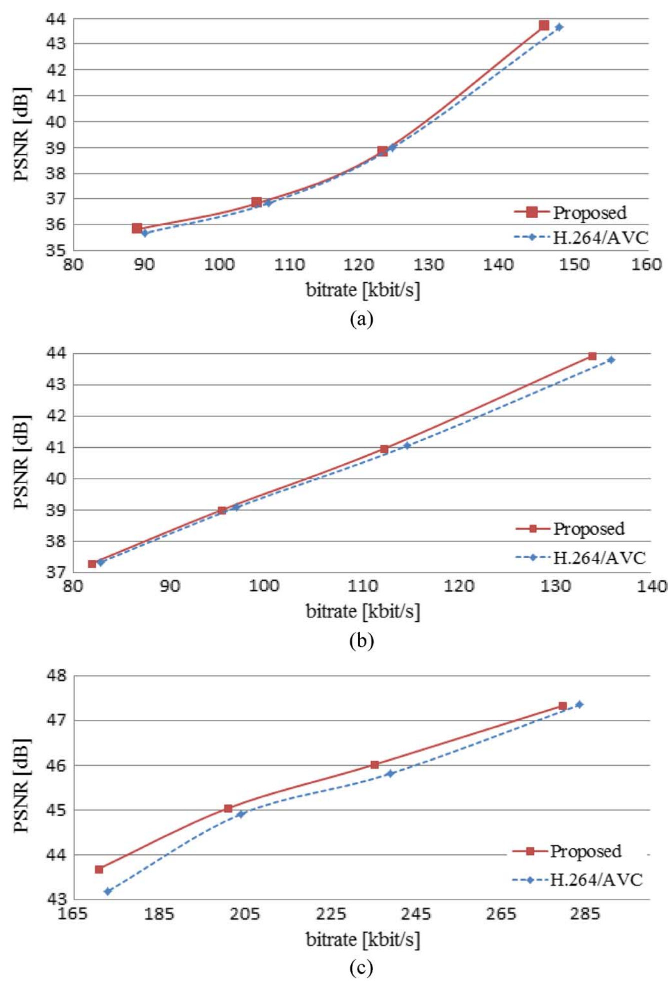


Fig. 16. R-D curves for the "Mobile" sequence: (a) result of 5th viewpoint depth video, (b) result of 7th viewpoint depth video, and (c) result of 6th synthesized viewpoint color video.

Results of the second experiment are presented in Table III, which show the contributions of the proposed method. From the table, we can see that the proposed method provides better coding performances than the original H.264/AVC encoder in terms of rate-distortion. In particular, the "Lampshade2" sequence showed the best performance since it contained simple boundaries, such as diagonal lines. Therefore, our method is seen to be efficient when the input depth contains sharp and simple boundaries. Another notable result is observed in the synthesized views. The coding performance for the synthesized views is much higher than that of the real depth video pairs. This fact indirectly indicates that the proposed method is valid for improving the rendering quality, and this effect becomes greater if the input depth data includes smaller depth errors such as the "Mobile" sequence, which is a computer graphic. Indeed, the coding gain for the synthesized view of the "Mobile" sequence was greater than 200% relative to the gain for its left or right viewpoint.

It should also be noted that the proposed method increased the computational complexity at both the encoder and decoder sides due to estimation of the partitioning information and production of partitioned intra prediction modes. Although we only explained the encoder side process, it must be noted that the decoder follows exactly the same process except for the number of partitioned intra prediction modes supposed to be produced. Only one partitioned intra prediction mode is produced from its mode information and PIPE type at the decoder side; therefore, the amounts of increased computational complexity are slightly smaller at the decoder side than at the encoder side. However, it becomes greater when we measure it as a term of run-time percentage difference defined in (4):

$$\Delta Time = \frac{Time_{proposed} - Time_{JM14.2}}{Time_{JM14.2}} \times 100\ [\%]. \quad (4)$$

Fig. 16 shows that the R-D curves for the "Mobile" sequence where two curves, the 5th and 7th viewpoint curves, converge at a low bitrate. This observation confirms that the proposed method is less efficient at extremely high quantization parameters (QPs) since geometric structures of previously encoded and reconstructed neighboring blocks are significantly degraded. However, in spite of the convergence of the two curves, the curve for the synthesized result shows a more stable coding gain due to the boundary effect [13].

Fig. 17 highlights the subjective quality of the synthesized views of the "Ballet" sequence. These results directly confirm

Fig. 17. Subjective quality comparisons for the 4th view of the "Ballet" sequence: (a) original 4th view, (b) the 4th view synthesized from the 3rd and 5th depth videos with no compression, (c) the 4th view synthesized when the stereo pair of depth videos is compressed by H.264/AVC, and (d) the 4th view synthesized when the stereo pair of depth videos is compressed using the proposed method.

that the proposed method noticeably contributes to an improvement in the subjective quality of the synthesis view, even if the coding gains are not as significant.

V. Conclusion

In this paper, we proposed an adaptive geometry-based intra prediction method for efficient depth video coding. The proposed method efficiently produced partitioned intra prediction modes around object boundaries, which reduced residual data related to data loss during the compression process. Our method originated from adaptive geometry-based block partitioning, and was optimized by utilizing the unique characteristics of depth video: smooth depth-level distribution within an object and sharp depth-level variation near object boundaries. Coding performance was then evaluated in terms of a bit-saving, quality of compressed depth video, and quality of synthesized view. The experimental results confirmed that the proposed method guarantees better coding performance for various types of depth data and improved subjective quality of synthesized views, but the proposed method has a trade-off between coding efficiency and computational complexity.

References

[1] A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauf, P. Eisert, and T. Wiegand, "3D video and free view-point video-technologies, applications and MPEG standard," in *Proc. IEEE Int. Conf. Multimedia and Expo (ICME 2006)*, Jul. 2006, no. 4037061, pp. 2161–2164.

[2] K. Mueller, A. Smolic, P. Merkle, M. Kautzner, and T. Wiegand, "Coding of 3D meshes and video textures for 3D video objects," in *Proc. Picture Coding Symp. (PCS 2004)*, Dec. 2004, pp. 15–17.

[3] C. Fehn, P. Kauff, M. Op de Beeck, F. Ernst, W. Ijsselsteijn, M. Pollefeys, L. Vangool, E. Ofek, and I. Sexton, "An evolutionary and optimized approach on 3D-TV," in *Proc. Int. Broadcast Conf.*, Sep. 2002, pp. 357–365.

[4] P. Merkle, A. Smolic, K. Mueller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *Proc. Int. Conf. Image Processing (ICIP 2006)*, Jan. 2006, vol. 1, pp. I201–I204.

[5] H. Oh and Y. S. Ho, "H.264 based depth map sequence coding using motion information of corresponding texture video," in *Proc. Pacific Rim Symp. Advances in Image and Video Technology (PSIVT 2006)*, Dec. 2006, pp. 898–907.

[6] S. T. Na, K. J. Oh, C. Lee, and Y. S. Ho, "Multi-view depth video coding method using depth view synthesis," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS 2008)*, 2008, pp. 1400–1403.

[7] Y. Morvan, D. Farin, and P. H. N. de With, "Depth-image compression based on an R-D optimized quadtree decomposition for the transmission of multiview images," in *Proc. Int. Conf. Image Processing (ICIP 2007)*, Oct. 2007, vol. 5, pp. 105–108.

[8] P. Merkle, Y. Morvan, A. Smolic, K. Muller, P. H. N. de With, and T. Wiegand, "The effect of depth compression on multiview rendering quality," in *Proc. 3DTV Conf. 2008*, May 2008, pp. 245–248.

[9] W. S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map coding with distortion estimation of rendered view," in *Proc. SPIE Visual Information Processing and Communication (VIPC 2010)*, Jan. 2010, vol. 7543, p. 75430B.

[10] C. Dai, O. D. Escoda, P. Yin, X. Li, and C. Gomila, "Geometry-adaptive block partitioning for Intra prediction in image and video coding," in *Proc. IEEE Int. Conf. Image Processing (ICIP2007)*, Sep. 2007.

[11] M. K. Kang, J. Lee, J. Y. Lee, and Y. S. Ho, "Geometry-based block partitioning for efficient intra prediction in depth video coding," in *Proc. SPIE Visual Information Processing and Communication (VIPC 2010)*, Jan. 2010, vol. 7543, p. 75430A.

[12] M. K. Kang, C. Lee, J. Y. Lee, and Y. S. Ho, "Adaptive geometry-based intra prediction for depth video coding," in *Proc. IEEE Int. Conf. Multimedia and Expo (ICME2010)*, Jul. 2010, pp. 1230–1235.

[13] K. J. Oh, S. Y. Kim, and Y. S. Ho, "Depth reconstruction filter for depth coding," *IET Electron. Lett.*, vol. 45, no. 6, pp. 1–2, Mar. 2009.

[14] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, pp. 11–15, Jan. 1972.

[15] ISO/IEC JTC1/SC29/WG11, Report on Experimental Framework for 3D Video Coding, N11631. Guangzhou, China, Oct. 2010.

[16] ISO/IEC JTC1/SC29/WG11, Applications and Requirements on 3D Video Coding, N11678. Guangzhou, China, Oct. 2010.

[17] ISO/IEC JTC1/SC29/WG11, Draft Call for Proposals on 3D Video Coding Technology, N11679. Guangzhou, China, Oct. 2010.

[18] Kakadu software. [Online]. Available: http://www.kakadusoftware.com/.

[19] Coding Results of Platelet-Based Encoder. [Online]. Available: http://vca.ele.tue.nl/demos/mvc/PlateletDepthCoding.tgz.

[20] MSR 3D Video Sequences. [Online]. Available: http://www.research.microsoft.com/vision/ImageBasedRealities/3DVideoDownload/.

[21] Joint Video Team, Reference Software Version 14.2. [Online]. Available: http://iphome.hhi.de/suehring/tml/download/old_jm/jm14.2.zip.

[22] Middlebury Stereo Datasets. [Online]. Available: http://vision.middlebury.edu/.

[23] ISO/IEC JTC1/SC29/WG11, Description of Exploration Experiments in 3D Video Coding, N11630. Guangzhou, China, Oct. 2010.

[24] ITU-T Q.6/16, Calculation of Average PSNR Differences Between RD-Curves, VCEG-M33, Mar. 2001.

[25] ISO/IEC JTC1/SC29/WG11, View Synthesis Algorithm in View Synthesis Reference Software 2.0 (VSRS2.0), M16090. Lausanne, Switzerland, Feb. 2009.

Min-Koo Kang (S'10) received the B.S. degree in electronic engineering from Inha University, Incheon, Korea, in 2008 and the M.S. degree in electrical engineering from Gwangju Institute of Science and Technology (GIST), Gwangju, Korea, in 2010. He is currently pursuing the Ph.D. degree at GIST.

His current research interests include digital image and video coding, multiview video and depth video coding, video quality assessment, and realistic broadcasting system.

Yo-Sung Ho (M'81–SM'06) received both B.S. and M.S. degrees in electronic engineering from Seoul National University, Seoul, Korea, in 1981 and 1983, respectively, and the Ph.D. degree in electrical and computer engineering from the University of California, Santa Barbara, in 1990.

He joined the Electronics and Telecommunications Research Institute (ETRI), Korea, in 1983. From 1990 to 1993, he was with Philips Laboratories, Briarcliff Manor, NY, where he was involved in development of the advanced digital high-definition television (AD-HDTV) system. In 1993, he rejoined the technical staff of ETRI and was involved in the development of the Korea DBS digital television and high-definition television systems. Since 1995, he has been with the Gwangju Institute of Science and Technology (GIST), Gwangju, Korea, where he is currently a professor in the School of Information and Communications. His research interests include digital image and video coding, image analysis and image restoration, advanced coding techniques, digital video, and audio broadcasting.