Object-Adaptive Depth Compensated Inter Prediction for Depth Video Coding in 3D Video System

Min-Koo Kang[a], Jaejoon Lee[b], Ilsoon Lim[b], Yo-Sung Ho[a]

[a]Gwangju Institute of Science and Technology (GIST)
261 Cheomdan-gwagiro, Buk-gu, Gwangju 500-712, Korea;

[b]Multimedia Lab. Samsung Electronics Co., Ltd.
San#14-1 Nongsea Dong, Giheung-gu Yongin-si Gyeonggi-do, 446-712 Korea

ABSTRACT

Nowadays, the 3D video system using the MVD (multi-view video plus depth) data format is being actively studied. The system has many advantages with respect to virtual view synthesis such as an auto-stereoscopic functionality, but compression of huge input data remains a problem. Therefore, efficient 3D data compression is extremely important in the system, and problems of low temporal consistency and viewpoint correlation should be resolved for efficient depth video coding. In this paper, we propose an object-adaptive depth compensated inter prediction method to resolve the problems where object-adaptive mean-depth difference between a current block, to be coded, and a reference block are compensated during inter prediction. In addition, unique properties of depth video are exploited to reduce side information required for signaling decoder to conduct the same process. To evaluate the coding performance, we have implemented the proposed method into MVC (multiview video coding) reference software, JMVC 8.2. Experimental results have demonstrated that our proposed method is especially efficient for depth videos estimated by DERS (depth estimation reference software) discussed in the MPEG 3DV coding group. The coding gain was up to 11.34% bit-saving, and it was even increased when we evaluated it on synthesized views of virtual viewpoints.

Keywords: 3D video coding, 3DTV, MVC, depth video coding, JMVC, inter prediction

1. INTRODUCTION

The MPEG (Moving Picture Experts Group) has been conducted various standardizations related to the 3D video system. One of the latest activities was about the FTV/3DV standard system for various 3DTV applications[1]. Nowadays, a depth-based representation such as MVD (multi-view video-plus-depth) or LDV[2] (layered depth video) has been mainly considered as an input of the FTV/3DV standard system. This system is being developed to support more realistic and immersive 3D feeling to viewers by providing a functionality of free viewpoint navigation based on virtual view synthesis technologies with depth data[3] and stereoscopic depth impression. To achieve this, many advanced technologies should be integrated into the system, called 3DV process chain, and balance studied[4].

Among all 3DV-realated technologies, 3D data compression is a key technology of the 3DV system because 3D data is usually captured by multiview camera setups, and as a result, it has huge amount of data. The 3D data mainly consist of color and depth videos. Currently, many efficient algorithms for multiview color video coding (MVC) have been proposed[5], but depth data compression is still remaining as one of the most troublesome processes in the 3DV process chain. There are various types of depth data as an input for the 3DV system; thus, it is hard to suggest a specific depth compression technique since characteristics of depth data are quite different each other according to their types and estimation methods. In this point of view, the MPEG 3D video (3DV) coding group is finalizing collecting test depth video to fairly conduct experiments of 3D video coding and start its standardization[6]. Therefore, an efficient depth video coding method based on the MPEG test sequences is strongly required and worth to study at this moment.

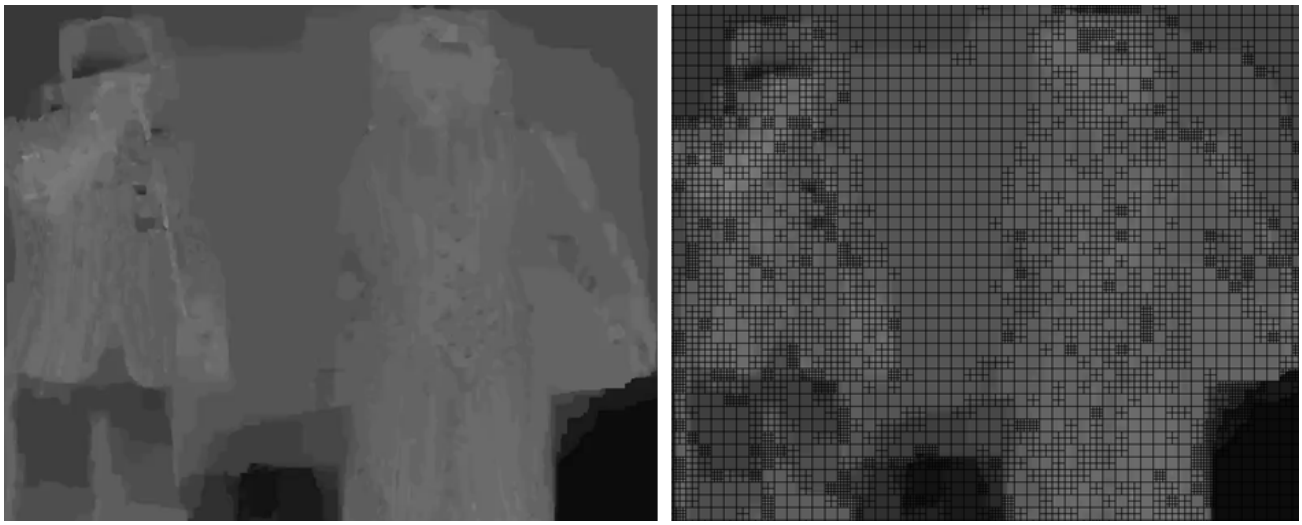
According to recent studies, approaches for efficiency of depth video coding can be classified into two groups. One group exploits correlation between color video and depth video, and share common information to reduce redundancy[7, 8]. The other group exploits only unique properties of depth video considering its role in the whole 3DV system, just supplement data for virtual view synthesis. The algorithms in the first group were proposed earlier than the second group.

In the time, many experts only focused on compression ratio, but did not much care about quality of a rendered view or system framework. As a result, those algorithms significantly depend on a coding order or a viewpoint structure of color and depth video, and those are valid for limited 3DV applications in spite of their significant coding efficiency. However, the second group, introduced later, is more generally available to various 3DV applications since their algorithms are independent with the experimental framework and exploiting only unique properties of depth video itself. For example, Kim *et al.* proposed a depth map coding method by distortion estimation of rendering view[9]. In their paper, they focused on the fact that depth data are only used for virtual view synthesis, and the boundaries of objects within a scene significantly affect quality of a rendered view. Another example is to exploit unique level distribution of depth video. Depth levels are homogeneous in an object, but they sharply change around object boundaries. Utilizing this property, Kang *et al.* proposed a geometry-based block partitioning for efficient intra prediction in depth video coding[10, 11].

In this paper, we propose another unique depth property-based method. The proposed method efficiently encodes test sequences of the MPEG 3DV coding group estimated by DERS (depth estimation reference software). The sequences have extraordinary depth variations between temporally successive frames and neighboring viewpoint frames. These make performance of inter prediction deteriorated and MVC (multiview video coding) structure is less efficient in the state of the art video codec. To solve these problems, we observed and analyzed unique characteristics of MPEG test sequences, and proposed depth compensated inter prediction method motivated by an IC (illumination compensation) technique[12].

2. ANALYSIS ON DEPTH VIDEO

Depth video has many different properties with that of color video. One of the most noticeable properties is its simple level distribution. Since each level in depth video indicates a distance between a camera and a position of an object within a captured scene, depth levels are all similar within an object, but there is abrupt level change around object boundaries. As a result of this property, most homogeneous regions are coded with the largest prediction mode (e.g. SKIP, inter 16x16, or intra 16x16), while boundary regions are coded with the smallest prediction mode (e.g. intra 4x4) under the conventional quadtree-based block partition structure of the state-of-the-art video codec[10, 11]. Figure 2 shows depth video of "Pantomime" sequence and its coded prediction blocks.

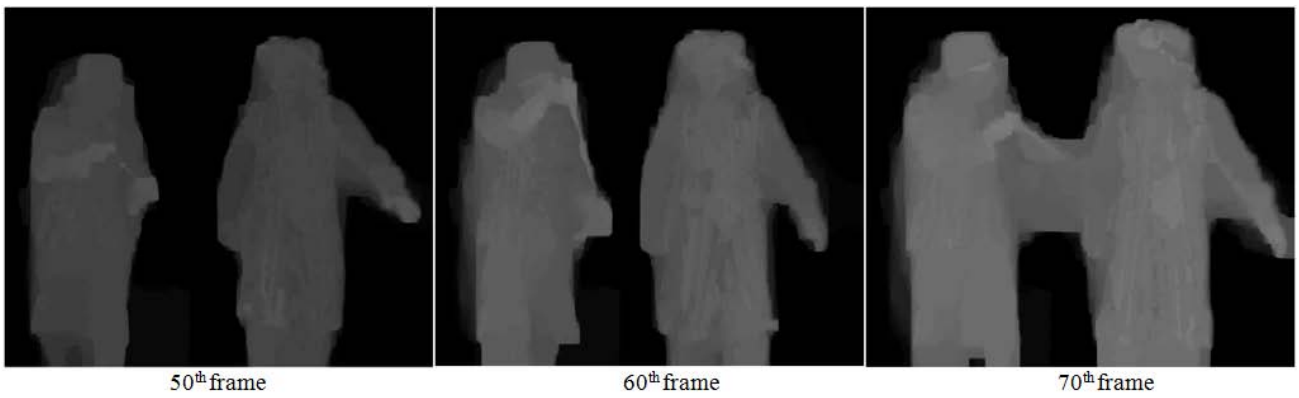


(a) (b)

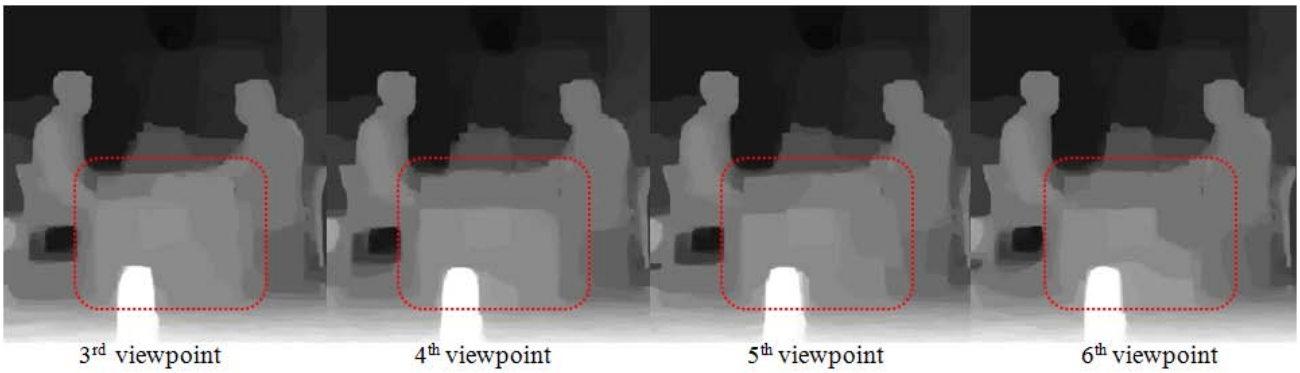
Figure 2. Depth video of "Pantomime" sequence: (a) encoded depth video, (b) coded prediction blocks

Another important depth video property, low temporal/spatial correlation, is shown in MPEG test sequences. This can be explained with an object motion toward z-axis and a methodological problem of depth estimation as depicted in Fig. 3 (a) and (b), respectively. Since depth is a distance between an object and a camera, depth clearly changes when the object moves toward z-axis and in different viewpoint as shown in Fig. 3 (a). New depth-level appearance makes motion estimation difficult although a current block, to be coded, and a corresponding reference block are closely located. In addition, depth estimation method implemented in DERS gives low depth-consistency between temporally successive

frames as well as spatially neighboring frames. For instance, depth values of the tables in Fig. 3 (b) should be the same because those are located at the same position, but those are incoherent.



(a)



(b)

Figure 3. Low temporal/spatial consistency of depth: (a) temporally successive frames of "Pantomime" sequence, (b) spatially neighboring viewpoints of "Laptop" sequence

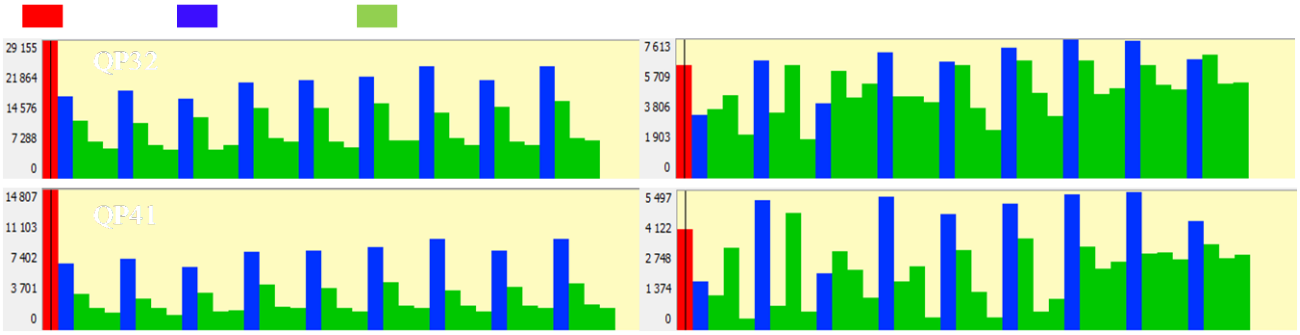
The state of the art video codec did not take this property into account. Therefore, inter prediction scheme is less efficient for depth video compression, and this makes the current video codec is sub-optimal for depth video although depth video has the same format with color video, YUV 4:2:0. Figure 4 (a) and (b) proves this where red-lines represent used motion vectors for inter prediction. According to Fig. 4, fewer regions are encoded by inter prediction with motion vectors in depth video than that of color video. In practice, P-picture and B-picture in the structure of hierarchical B-picture coding do not always guarantee better depth coding performance with respect to bit-saving as shown in Fig. 5. Similarly, P-viewpoint and B-viewpoint in MVC structure do not always guarantee better depth coding performance as shown in Fig. 6. Usually, MVC structure guarantees superior coding performance to the simulcast coding, but this is almost opposite in depth video coding because of the above reasons. Therefore, a technique to increase temporal/spatial correlation is strongly required for further improvement of depth video coding efficiency.

3. RELATED METHOD

It intuitionally looks easy to compensated depth variation induced by the reasons mentioned in section 2 because of its simple level distribution. Just simple addition or subtraction operation might be adequate to compensate a depth difference between a current block and a reference block. In this point of view, we introduce a related work proposed by Kim *et al*[12] before we explain the proposed method for better understanding.

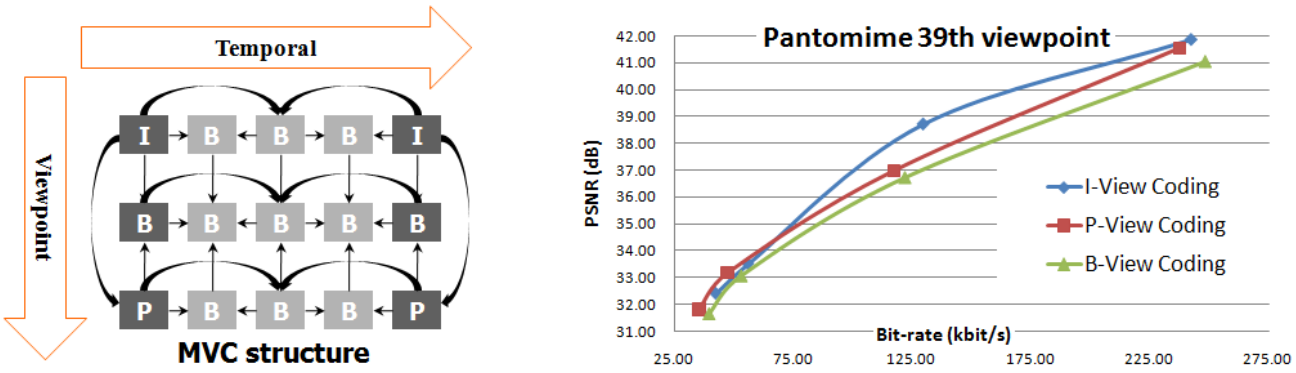
(a) (b)

Figure 4. Motion vector comparison of "Pantomime" sequence: (a) color video, (b) depth video



(a) (b)

Figure 5. Bitrate comparison of "Pantomime" sequence under hierarchical B-picture coding structure: (a) bitrate of color video, (b) bitrate of depth video



(a) (b)

Figure 6. Bitrate comparison of "Pantomime" sequence under MVC structure: (a) structure of multiview video codin, (b) rate-distortion curves of three viewpoint coding scenarios

Most videos captured by a multiview video setup include many inconsistent problems of video features in spite of producer's careful attention. One of the most representative problems is illumination change between videos captured by different viewpoints. This difference results in low compression efficiency, and illumination change-adaptive motion compensation (ICA MC) was proposed to solve this problem.

In this method, illumination compensated inter 16x16 mode competes with other prediction modes with respect to rate-distortion, and it is selected as the best prediction mode if it gives the lowest rate-distortion cost as depicted in Fig. 7. First, MR_SAD (mean-removed SAD) such as Eq. (4) is used instead of the conventional SAD such as Eq. (1) to conduct ICA motion estimation because illumination change is regarded as a mean difference of MBs. In Eq. (4), M_{cur}

and M_{ref} and represent mean of a current block and a reference block, and $f(i, j)$ and $r(i, j)$ represent a pixel intensity of a current slice and a reference slice, respectively. After the motion estimation, different value of illumination change (DVIC) is calculated by following Eq. (5), and this value is used to reconstruct a illumination compensated inter prediction. Finally, a difference between the DVIC and a predicted DVIC from neighboring MBs, defined as dpcm_of_dvic, is encoded and transmitted to a decoder. The following Fig. 7 depicts the decoder structure of ICA MC.

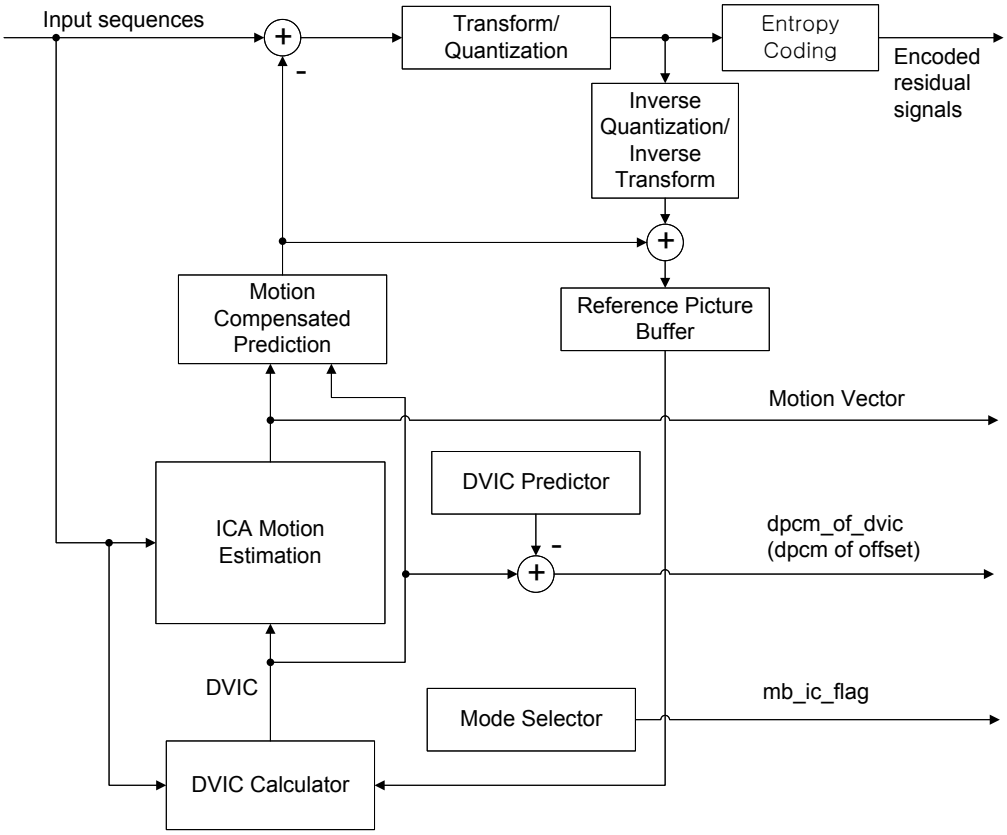


Figure 6. Encoder block diagram of the MB-based adaptive illumination change compensation method

$$SAD(x,y) = \sum_{i=m}^{m+S-1} \sum_{j=n}^{n+T-1} \left| f(i,j) - r(i+x, j+y) \right| \tag{1}$$

$$M_{cur} = \frac{1}{S \times T} \sum_{i=m}^{m+S-1} \sum_{j=n}^{n+T-1} f(i,j) \tag{2}$$

$$M_{ref}(p,q) = \frac{1}{S \times T} \sum_{i=p}^{p+S-1} \sum_{j=q}^{q+T-1} r(i,j) \tag{3}$$

$$MR\_SAD(x,y) = \sum_{i=m}^{m+S-1} \sum_{j=n}^{n+T-1} \left| \{ f(i,j) - M_{cur} \} - \{ r(i+x, j+y) - M_{ref}(m+x, n+y) \} \right| \tag{4}$$

$$DVIC = M_{cur} - M_{ref} \tag{5}$$

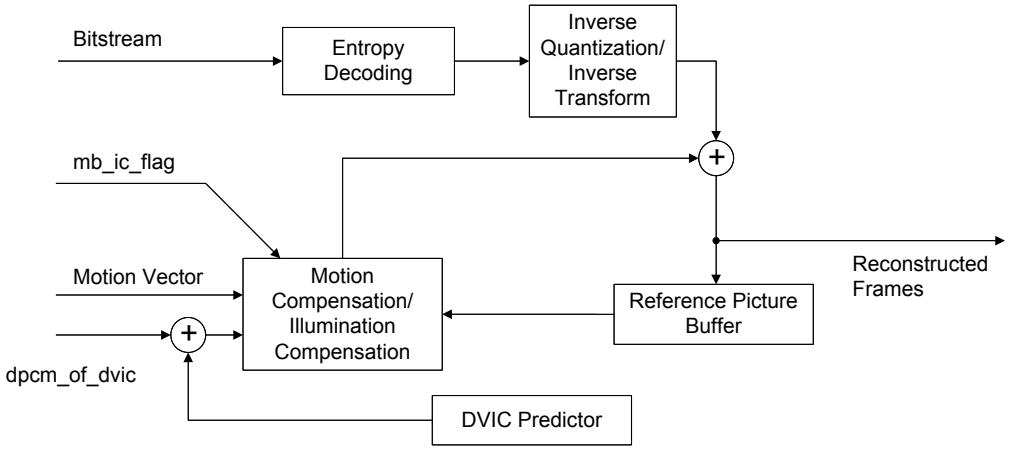


Figure 7. Decoder side block diagram of the MB-based adaptive illumination change compensation method

Depth variation induced by the reason in section 2 can be treated similarly, but depth variation needs one more consideration. Depth variation changes with different intensities if there is superimposition of objects while illumination change is always the same within a MB. Therefore, compensation of depth variation should be adaptively treated by considering object boundaries

4. PROPOSED METHOD

The depth variation induced by the above two causes can be compensated by adding or subtracting a constant corresponding to a depth difference during motion estimation and compensation. In this point of view, IC (illumination compensation) technique[5] of MVC can be applied for depth video coding, but existence of superimposition of objects within a block makes the problem complicated because there might be different depth variation according to each object if there exist more than two objects. Therefore, we propose a new method that separately compensates each depth variation according to each object within a block during inter prediction. The proposed method basically exploited unique properties of depth video: homogeneous level distribution in an object, and sharp level change around boundaries. These provide us convenience to develop the proposed method without side information to inform compensation to decoder. The following Fig. 8 (a) depicts the block diagram of the proposed motion estimation (ME) / motion and object-adaptive depth compensation (MOADC) encoder, and Fig. 8 (b) shows its flowchart from template access to final prediction mode generation.

4.1 OBJECT-ADAPTIVE DEPTH COMPENSATION

According to analysis of MPEG's test depth sequences, depth variation about each corresponding pixel shows a mean difference of two corresponding blocks. Therefore, a method using MR-SAD and compensating an offset is available for depth video compression to improve performance of inter prediction similarly to ICA ME/MC method. Figure 9 depicts concept of MR_SAD, and it shows MR_SAD gives lower distortion at the step of motion estimation than the conventional SAD where a mean difference is calculated and compensated at the step of predicted block generation. As a result, the correct reference block can be selected as the best prediction block in spite of a significant depth variation by using MR_SAD measure instead of the conventional SAD.

The problem occurs when a current block located at an object boundary region. Levels of depth variation according to each region, foreground or background, are different because depth is distance information, and each depth variation is applied with an object unity. Thus, we need to separate a block into sub-regions, foreground and back ground, before we conduction motion estimation using MR-SAD. After this, an object-adaptive MR_SAD can be conducted as show in Fig.

10. This approach makes a current block possible to find an adequate corresponding reference block wherever the current block located, but we need to calculate each depth offset according to each region, and compensate it separately.

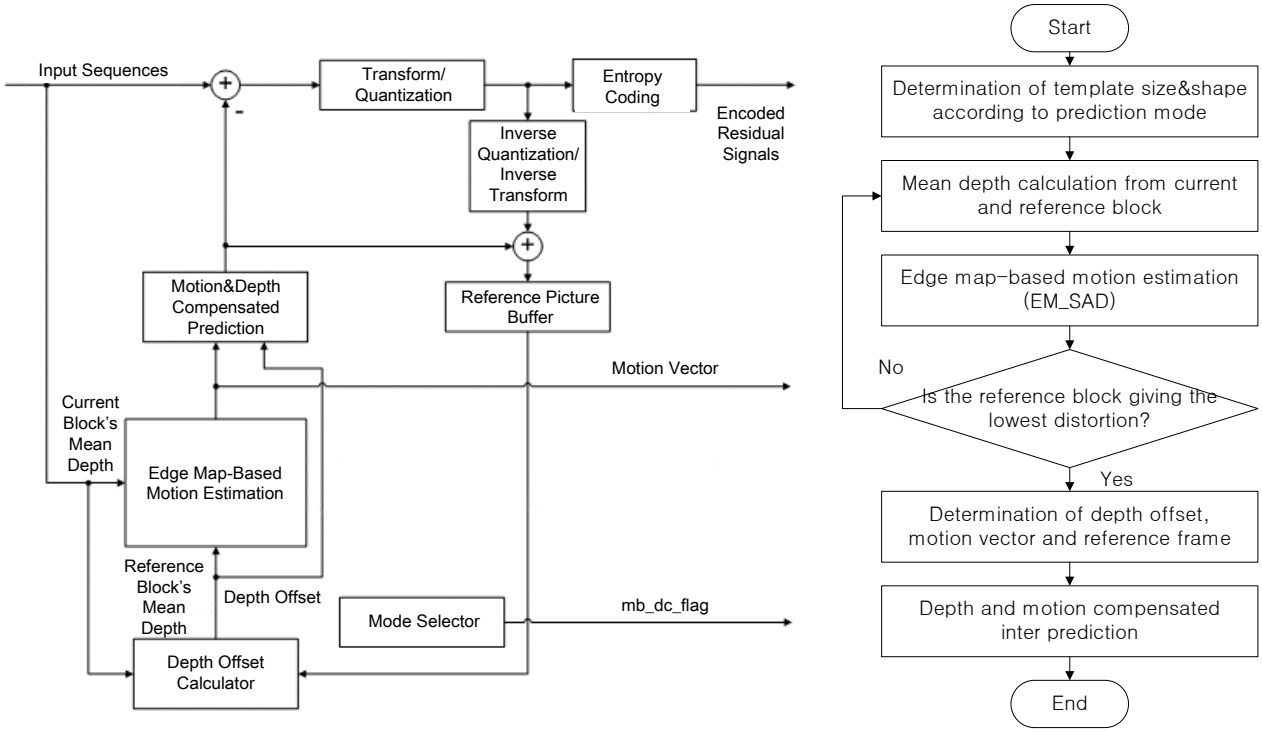


Figure 8. Proposed object-adaptive depth compensated inter prediction algorithm: (a) block diagram, (b) flowchart

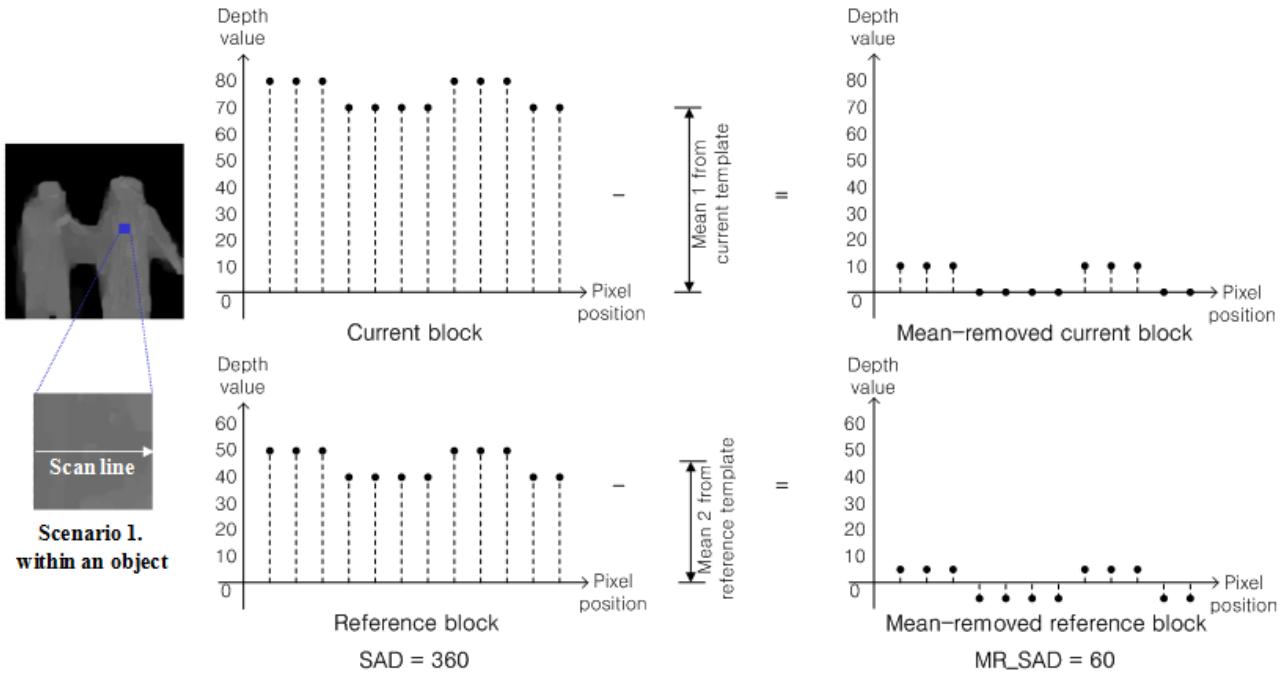


Figure 9. Distortion measure comparison when corresponding blocks are located inside an object

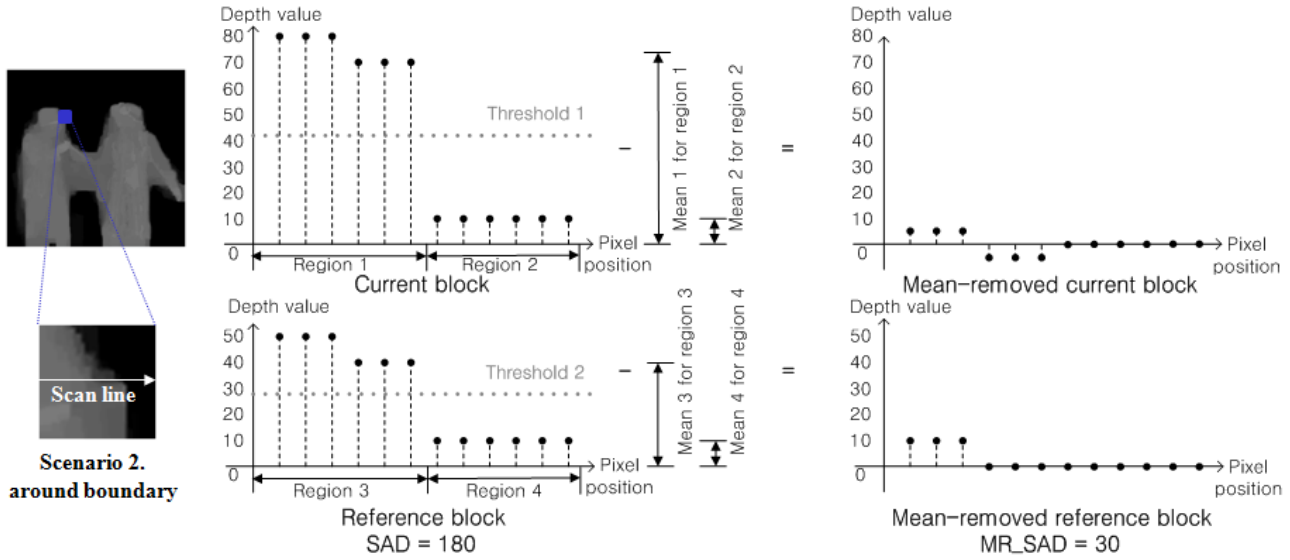
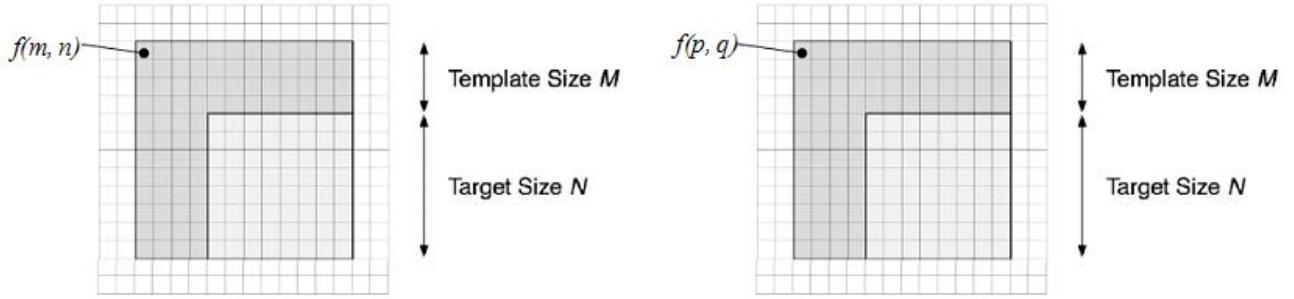


Figure 10. Distortion measure comparison when corresponding blocks located around an object boundary

4.2 TEMPLATE DIVISION AND DEPTH OFFSET CALCULATION

To conduct a block separation and depth offsets calculation, we need to access pixels in corresponding blocks, and signal related side information. However, exploiting the homogeneous property of depth data, we access templates instead of pixels in corresponding blocks where template means a set of neighboring pixels previously coded and decode. The use of template makes decoder possible to conduct the same processes done in encoder without the side information, but appropriate template size and shape should be determined beforehand to make depth offsets from a pair of templates the same with depth offsets from a pair of blocks. Figure 10 depicts templates of a current block and a reference block when the blocks do not include an object boundary where *f(m, n)* and *f(p, q)* represent levels of top-left pixels of the current block and the reference block, respectively. Eq. (6) and Eq. (7) explains how to calculate each mean depth of each template when the blocks are given like Fig. 11, and a depth offset is calculated like Eq. (9).



(a) (b)

Figure 11. Target block size and neighboring template size: (a) current block (b) reference block

$$M_{CT} = \frac{1}{NPT}\left[\sum_{i=m}^{m+M+N-1}\sum_{j=n}^{n+M-1} f(i,j) + \sum_{i=m}^{m+M-1}\sum_{j=n+M}^{n+M+N-1} f(i,j)\right] \tag{6}$$

$$M_{RT} = \frac{1}{NPT}\left[\sum_{i=p}^{p+M+N-1}\sum_{j=q}^{q+M-1} r(i,j) + \sum_{i=p}^{p+M-1}\sum_{j=q+M}^{q+M+N-1} r(i,j)\right] \tag{7}$$

$$The Number\ of\ Pixels\ in\ the\ Template\ (NPT) = 2 \times N \times M + M^2 \tag{8}$$

$$Depth\ Offset = M_{CT} - M_{RT} \tag{9}$$

Figure 12 depicts templates division and its flowchart when the blocks include an object boundary. First, maximum depth and minimum depth are detected in a template, and then an initial threshold is calculated by taking an average of the maximum depth and the minimum depth. After this, the template is separated into two regions, foreground and background. If intensity of a pixel is greater than the threshold, it belongs to foreground. If intensity of a pixel is lesser than the threshold, it belongs to background. Then, mean depth of each region is calculated to update the threshold. A new threshold is an average of mean depth of foreground and back ground. This cycle is repeated until there is no change on an updated threshold. Then, final template division and mean depth calculation are conducted using the fixed threshold. This process is applied for both a current block and a reference block, and each depth offset is independently calculated according to each divided template as shown in Eq. (10).

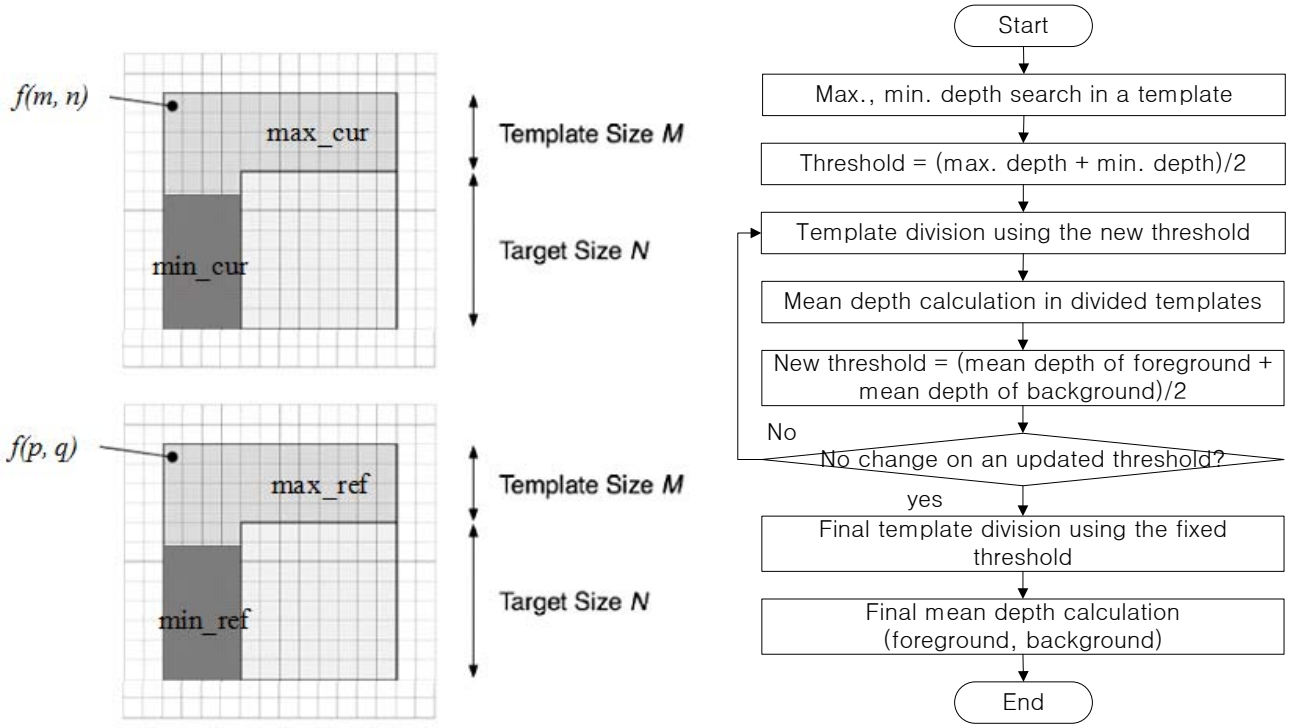


Figure 12. Target block size and neighboring template size: (a) current block (b) reference block

4.3 EDGE MAP-BASED MOTION ESTIMATION

During ME, a wrong reference block can be selected as the best reference block when all intensities are the same within an object by the proposed method. For example, if mean depth of divided templates exactly stand for mean depth of divided blocks, and all intensities within each divided block are the same, than mean-removed blocks have all zero intensities. As a result, MR_SAD gives zero distortion in the condition, and a wrong reference block is selected shown in Fig. 13 (a). Then, decoder integrated in encoder side produces an object-adaptive depth compensated inter prediction block using the estimated motion vector and the calculated offsets, and the prediction block results in an unwanted residual block because of the different boundary orientation shown in Fig. 13 (b).

Therefore, finding a correct shape and orientation of a boundary is most important in the proposed method during motion estimation. To achieve this, we utilized edge map since an edge map emphasizes a boundary and eliminates a mean level.

Thus, edge map-based SAD is calculated instead of the normal MR_SAD about the original blocks to improve accuracy of motion estimation. In this manner, a correctly shaped reference block is selected, and then, a correct MOADC is conducted using the reference block and previously calculated offsets. Figure 14 depicts the block diagram of the proposed method.

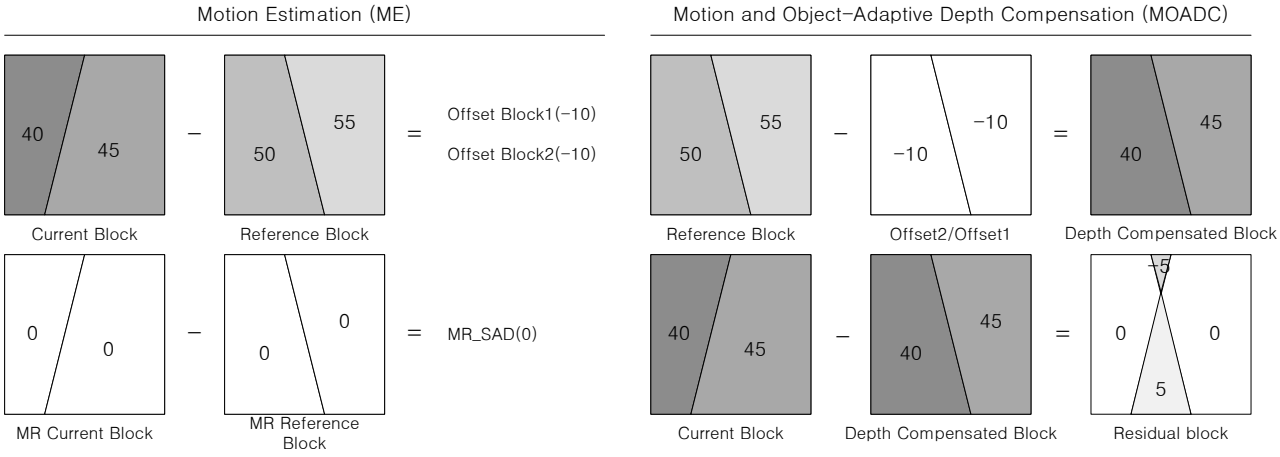


Figure 13. Problem of MR_SAD in a certain condition: (a) ME step (b) MOADC step

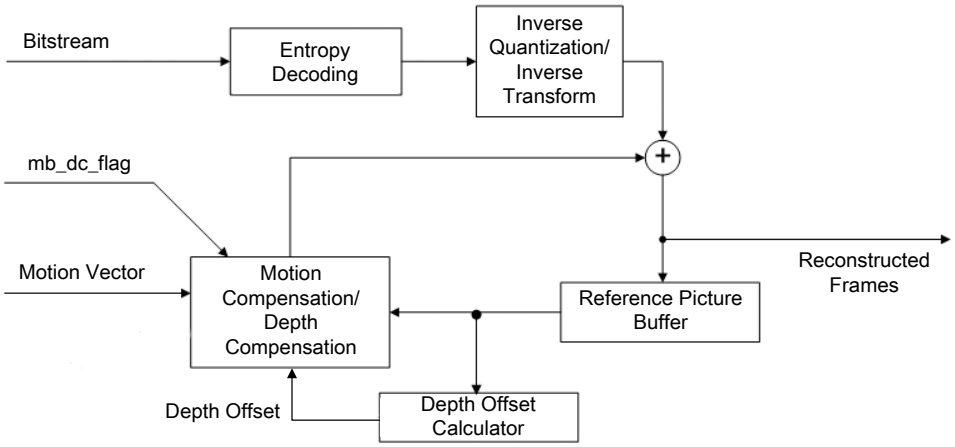


Figure 14. Decoder side block diagram of the proposed object-adaptive depth compensated inter prediction method

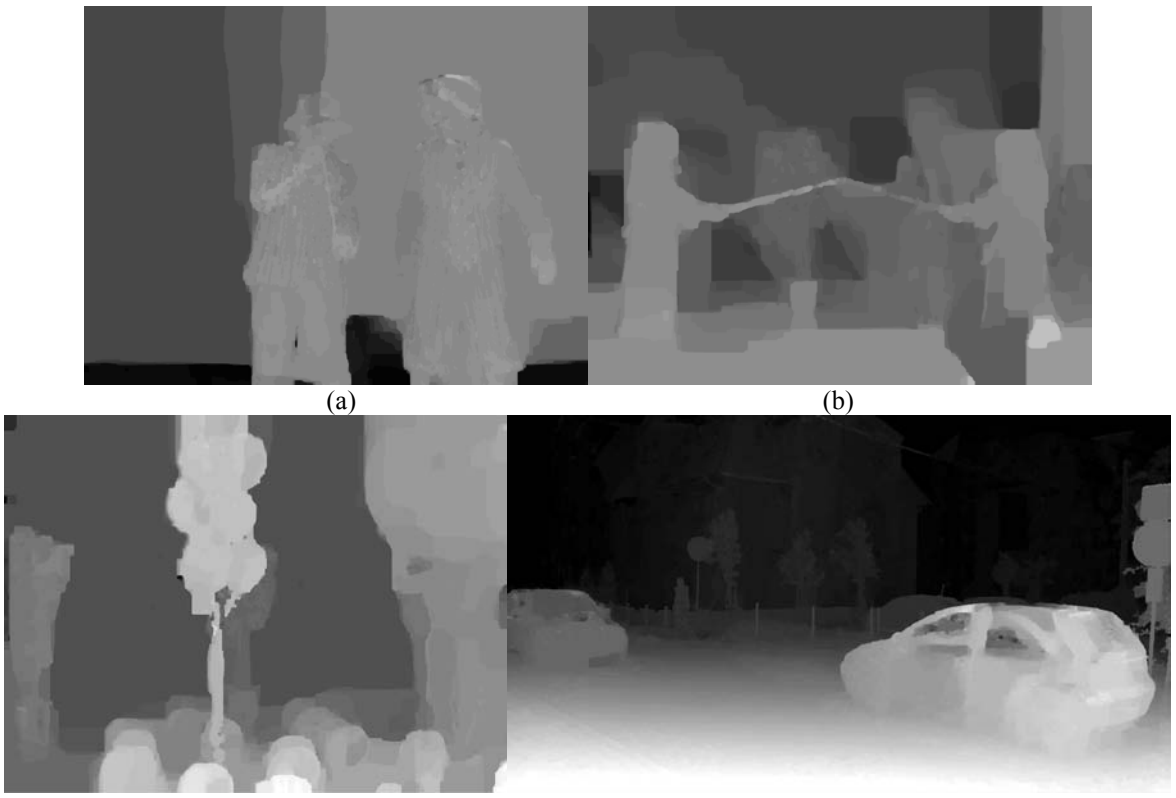
5. EXPERIMENTAL RESULTS

We have extended the direct and inter 16x16 prediction schemes of JMVC 8.2 with the proposed method, and compared it with the original JMVC 8.2. Experimental coding conditions followed the experimental framework with 3-view configuration of the MPEG 3DV coding group [6]. Experiments were conducted with various MPEG test sequences with three viewpoints color and depth video pairs. For evaluation of temporal/spatial inter prediction performance, a MVC structure, I-B-P picture coding, and hierarchical-B picture coding were applied. In the 3D video system, not only depth video compression itself but also a rendered virtual view is important; thus, a virtual view was synthesized using left and right color and depth video pairs, and the synthesized views with or without compression were compared to evaluate

coding performance. In addition, we have used relatively higher QP values, from 39 to 46, for depth video compression. These QPs are practical because the rendering quality is more sensitive to quality of color video; thus depth is coded with relatively higher QPs than that of color video. Moreover, this range clearly shows degradation of rendering quality. Therefore, it makes easy to evaluate the coding efficiency of the proposed method on a synthesized view. Table 1 summarizes used test sequences and common coding conditions, and Fig. 15 shows captured images of test sequences.

Table 1. MPEG 3DV test sequences description and common test condition

Name	Image Property	Used Viewpoints	Camera Arrangement
Pantomime	1280x960, 30fps, 300 frame	37, 39, 41	80 cameras with 5cm spacing
Kendo	1024x768, 30fps, 300 frame	1, 3, 5	7 cameras with 5cm spacing, moving camera
Balloons	1024x768, 30fps, 300 frame	1, 3, 5	7 cameras with 5cm spacing, moving camera
Poznan_Street	1920x1088, 25fps, 250frame	3, 4, 5	9 cameras with 13.75cm spacing
Common Condition	I-B-P MVC structure, GOP = 15 with hierarchical B-picture coding, QP(34, 39, 44, 46), CABAC, search range = 6		



(a) (b) (c) (d)

Figure 15. Captured images of the test sequence: (a) Pantomime, (b) Kendo, (c) Balloons, (d) Poznan_Street

Table 2 shows experimental results in terms of BDBR (Bjonteggard Delta BitRate) and BDPSNR (Bjonteggard Delta PSNR)[13] although these terms do not exactly stand for depth coding performance because those are designed for color video coding. In the table, three viewpoint videos of each test sequence were compressed as I-view, P-view, and B-view in 3-view configuration of MVC structures. The proposed method showed significant coding efficiency with respect to bit-saving, and the coding efficiency becomes greater when more neighboring views are considered as reference frames for inter prediction. For example, the coding performance of B-View showed greater than twice of I-View. The

maximum coding performance was up to 11.34% with respect to bit-saving, but it was not stable among all test sequences.

The synthesized results were calculated from the summation bit-rate of a pair of left and right depth videos and quality of a synthesized view. The proposed method showed better coding performance about synthesized views, too. Bitrates for depth video compression were significantly reduced while quality of synthesized views is preserved. The maximum coding gain was shown in the 39th virtual viewpoint video of "Pantomime" sequence. It is about 16% bit-saving as shown in Table 3.

Table 2. Experimental results on depth video (PSNR, Bitrate)

Viewpoint		**I-View**		**B-view**		**P-View**	
Measure		**BDPSNR (ΔdB)**	**BDBR (Δ%)**	**BDPSNR (ΔdB)**	**BDBR (Δ%)**	**BDPSNR (ΔdB)**	**BDBR (Δ%)**
Sequence	Pantomime	0.48	-8.21	0.28	-4.51	0.38	-8.23
	Kendo	0.30	-5.88	0.41	-5.94	0.35	-6.10
	Balloons	0.16	-3.99	0.36	-7.00	0.10	-2.72
	Poznan_Street	-0.26	6.59	0.35	-10.08	0.48	-11.34
	Average	0.17	-2.87	0.35	-6.88	0.33	7.10

Table 3. Experimental results on synthesized video(PSNR, Bitrate)

	Pantomime 39th virtual viewpoint video (synthesized)					
IC	**Off**		**On**		**Gain**	
QP	**Bitrate[kbps]**	**PSNR[dB]**	**Bitrate[kbps]**	**PSNR[dB]**	**BDPSNR(ΔdB)**	**BDBR(Δ%)**
34	552.61	35.09	466.31	35.12	0.27	-16.18
39	289.51	33.88	239.07	33.89		
44	126.54	32.70	108.71	32.71		
46	93.68	32.44	82.55	32.37		

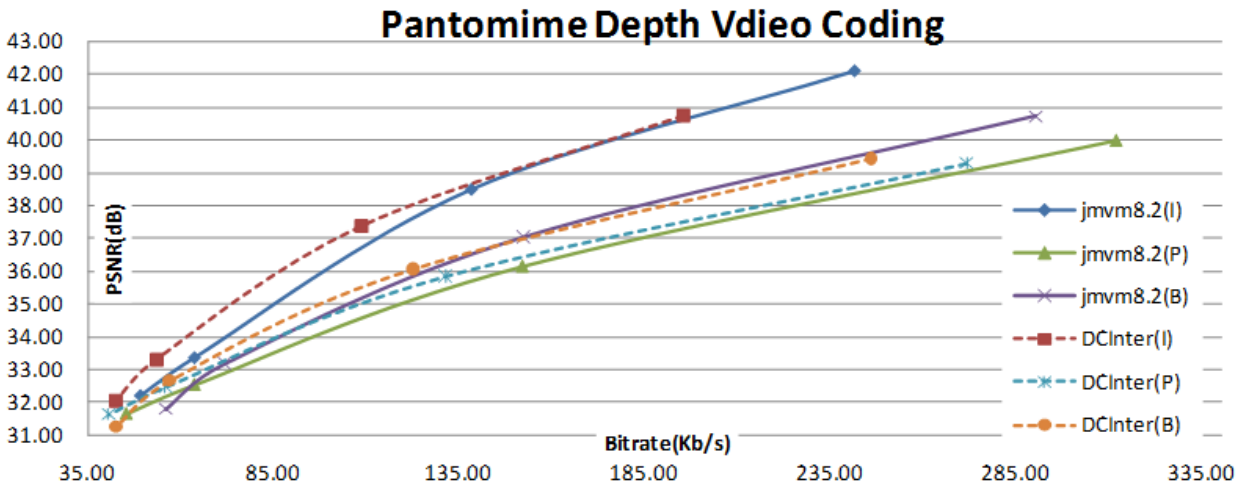


Figure 16. R-D curve of "Pantomime" depth video

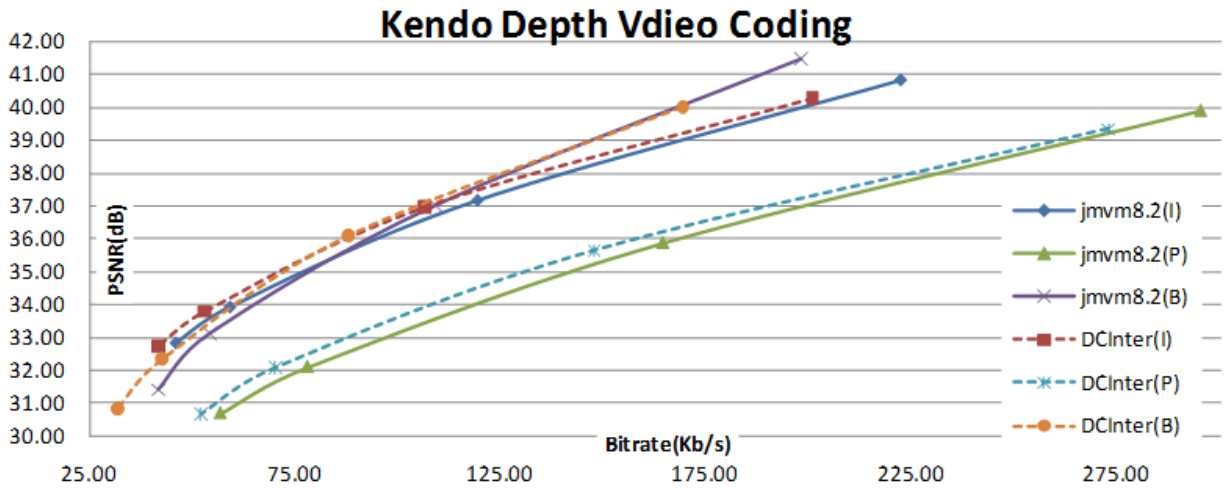


Figure 17. R-D curve of "Kendo" depth video

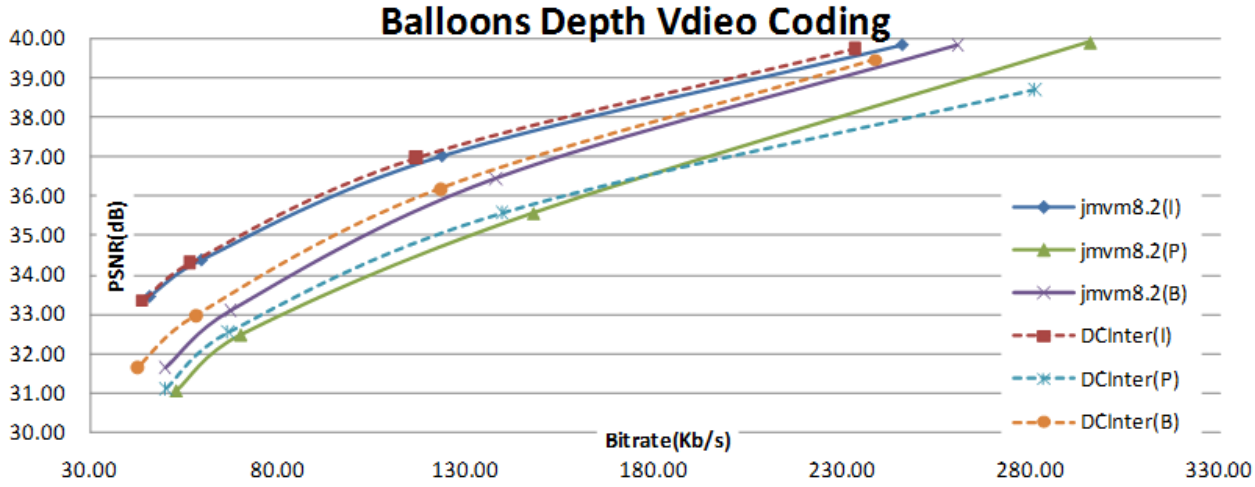


Figure 18. R-D curve of "Balloons" depth video

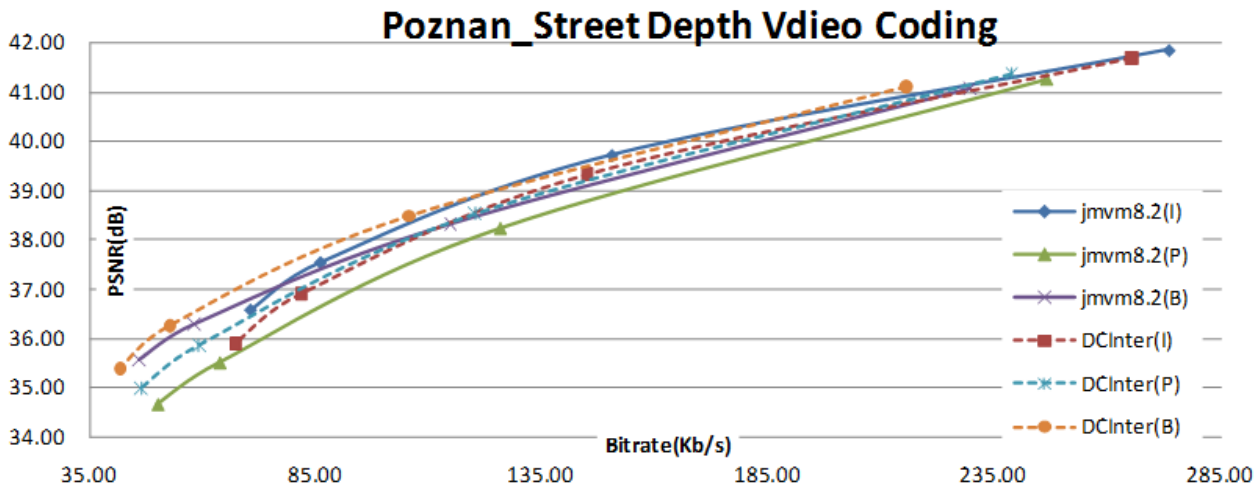


Figure 19. R-D curve of "Poznan_Street" depth video

6. CONCLUSION

In this paper, we have proposed an object-adaptive depth compensated inter prediction for depth video coding in the 3D video system. Since extraordinary depth variation, found in MPEG test depth sequences, results in a drop of inter prediction efficiency in the conventional MVC structure, we proposed an object-adaptive depth compensation technique to increase temporal/spatial correlation during motion estimation and motion compensation for inter prediction. The proposed method can be implemented into any current video codec; thus, it is applicable to various 3D video applications such as stereoscopic video or auto-stereoscopic video. The experimental results showed that the coding efficiency was significantly improved with respect to bit-saving, and quality of synthesized view was preserved. In this paper, we only extended direct and inter 16x16 modes, but the proposed method will be implemented into various inter prediction modes in the future. Moreover, a more reliable size and shape of template will be designed according to each inter prediction mode to further improve coding efficiency.

ACKNOWLEDGEMENTS

This work was supported in part by Samsung Electronics, Co., Ltd.

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," IEEE International Conference on Multimedia and Expo, No. 4037061, pp. 2161-2164, July 2006.
2. P. Merkle, A. Smolic, K. Müller, and T. Wiegand, “Multi-view video plus depth representation and coding,” IEEE International Conference on Image Processing, No. 1, pp. 1201-1204, Jan. 2006.
3. S. C. Chan, H. Y. Shum, K. T. Ng, “Image-Based Rendering and Synthesis," IEEE Signal Processing Magazine, Vol. 24, Issue 6, 2007.
4. ISO/IEC JTC1/SC29/WG11, “Applications and Requirements on 3D Video Coding,” N11678, Oct. 2010.
5. ISO/IEC JTC1/SC29/WG11, “Survey of algorithms used for multi-view video coding (MVC)," N6909, Jan. 2005.
6. ISO/IEC JTC1/SC29/WG11, “Description of Exploration Experiments in 3D Video Coding,” N11630, Oct. 2010.
7. S. T. Na, K. J. Oh, C. Lee, and Y. S. Ho, “Multi-view depth video coding method using depth view synthesis,” IEEE International Symposium on Circuits and Systems, pp. 1400-1403, 2008.
8. H. Oh and Y. S. Ho, “H.264-based depth map sequence coding using motion information of corresponding texture video,” Rim Symposium on Advances in Image and Video Technology, pp. 898-907, Dec. 2006.
9. W. S. Kim, A. Ortega, P. Lai, D. Tina, and C. Gomila “Depth map coding with distortion estimation of rendered view,” SPIE Visual Information Processing and Communication, Vol. 7543, pp. 75430B, Jan. 2010.
10. M. K. Kang, J. Lee, J. Y. Lee, and Y. S. Ho, “Geometry-based block partitioning for efficient intra prediction in depth video coding," SPIE Visual Information Processing and Communication, Vol. 7543, pp. 75430A, Jan. 2010.
11. M. K. Kang, C. Lee, J. Y. Lee, and Y. S. Ho, "Adaptive Geometry-based Intra Prediction for Depth Video Coding," IEEE International Conference on Multimedia & Expo (ICME 2010), pp. 1230-1235, July 2010.
12. J. H. Hur, S. Cho, and Y. L. Lee, “Adaptive Local illumination change compensation method for H.264/AVC multiview video coding,” IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, No. 11, pp. 1496-1505, 2007.
13. ITU-T Q.6/16, “Calculation of average PSNR differences between RD-curves,” VCEG-M33, March 2001.