Temporally Consistent Depth Map Estimation for 3D Video Generation and Coding

Sang-Beom Lee, Yo-Sung Ho*

Gwangju Institute of Science and Technology (GIST), Gwangju 500-712, Korea

Abstract: In this paper, we propose a new algorithm for temporally consistent depth map estimation to generate three-dimensional video. The proposed algorithm adaptively computes the matching cost using a temporal weighting function, which is obtained by block-based moving object detection and motion estimation with variable block sizes. Experimental results show that the proposed algorithm improves the temporal consistency of the depth video and reduces by about 38% both the flickering artefact in the synthesized view and the number of coding bits for depth video coding.

Key words: three-dimensional television; multi-view video; depth estimation; temporal consistency; temporal weighting function

I. INTRODUCTION

Due to great advancements in computing power, interactive computer graphics, digital transmission and immersive displays, we can experience and reproduce simulations of reality [1-2]. Technological advances in displays have been focused on improving the range of vision and immersive feelings, such as wide-screen displays, high-definition displays, immersive displays and 3D displays. Recently, Three-Dimensional Television (3DTV) has been in the spotlight as one of the next-generation broadcasting services [3-4]. Due to advances in 3D displays, 3DTV can now provide users with a feeling of presence from the simulation of reality [5]. In the near future, we expect 3DTV to be realized including 3D content generation, coding, transmission and display.

Figure 1 shows a conceptual framework of a 3D video system. A 3D video system includes the entire process of acquisition, image processing, transmission and rendering by means of 3D video including *N*-view colour and depth images. A 3D video is produced by 3D cameras, such as stereoscopic cameras, depth cameras or multi-view cameras. Depth cameras allow direct acquisition of depth data; otherwise, depth in the video is generated by depth estimation algorithms. A 3D video can be rendered by various types of display systems, such as stereoscopic displays, *M*-view 3D displays or even 2D displays. It can be compatible with a conventional 2D display by selecting a single viewing angle according to user preference.

Given the increasing diversity of 3D services and display systems, proper rendering techniques for 3D video are required, particularly for multi-view video. If the number of views, *N*, in the multi-view video is fewer than that of the input viewpoints, *M*, of the 3D display system, rendering is impossible. Furthermore, if the distance between multi-view cameras is too large, viewers may feel visual fatigue — the recommended camera baseline is 50-80 mm; thus, intermediate view synthesis is necessary. Intermediate views are synthesized images generated at intermediate virtual viewpoints between 2 real cameras. Natural rendering of 3D video is beneficial in reducing eye discomfort.

Received: 2013-02-16
Revised: 2013-04-01
Editor: ZHU Xiuchang

A new algorithm is proposed to generate the temporally consistent depth map. We propose a depth estimation method that adaptively computes the matching cost using a temporal weighting function.

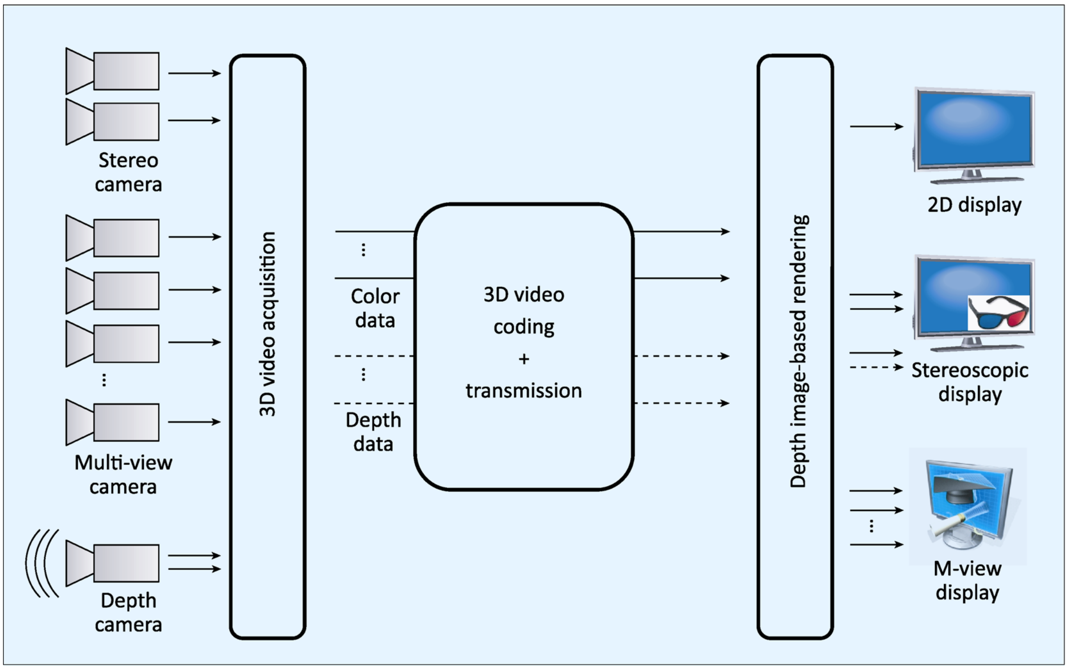


Fig.1 *3D video system*

To synthesize intermediate views from virtual viewpoints, depth information is required. Many studies have been conducted on depth acquisition [6-9]. However, the output depth maps have numerous errors due to the inherent sensor problem, boundary mismatch and occlusion/disocclusion. Particularly, because the depth map is estimated independently for each frame, depth values fluctuate. In other words, we notice inconsistent depth values in static regions as the frame changes. This temporal inconsistency problem may cause flickering artefacts in the synthesized views.

In the present paper, we propose a new algorithm for improving temporal consistency of the depth in the video. Specifically, we add a temporal weight to the existing matching function using a motion-compensated depth map. To obtain the reconstructed depth map, we detect moving objects and estimate motion using variable block sizes. Motion estimation and motion compensation are performed between the current and previous images.

The present paper is organized as follows. In Section II, we introduce related works in detail. We represent the proposed algorithm for temporally consistent depth estimation in Section III. After evaluating and analysing the proposed algorithm with several experiments in Section IV, we draw the conclusion in Section V.

II. RELATED WORKS

2.1 Disparity-depth relationship

Figure 2 illustrates the disparity-depth relationship. Let us assume that two pinhole cameras are located at C_l and C_r, and the optical axes of these two cameras are parallel to the z-axis. In addition, we assume that a certain 3D point is located at P and projected onto (x_l, y) on the left image plane and (x_r, y) on the right image plane. Then, the relationship between the disparity value d and the depth value Z can be defined by:

$$Z = \frac{Bf}{x_l - x_r} = \frac{Bf}{d} \quad (1)$$

where B and f represent the camera baseline and the focal length, respectively. Eq. (1) shows that the disparity estimation process will determine the real depth value.

2.2 Disparity computation using multi-view images

In general, the multi-view depth estimation

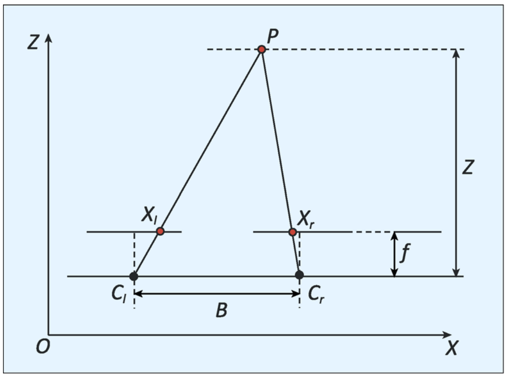


Fig.2 *Disparity-depth relationship*

algorithm consists of three main steps: disparity computation, error minimization and disparity-to-depth conversion.

The first step is to compute the matching cost for each pixel in the target view. Since there are three or more views in the multi-view images, the disparity can be calculated by comparing the target view to the left and right views simultaneously. The matching function is defined by:

$$E(x,y,d) = E_{\text{data}}(x,y,d) + \lambda_s E_{\text{smooth}}(x,y,d) \tag{2}$$

where d and λ_s denote disparity candidates and the weighting factor for $E_{\text{smooth}}(x,y,d)$, respectively. $E_{\text{data}}(x,y,d)$ represents the difference between intensities in the target and reference views. $E_{\text{data}}(x,y,d)$ is defined by:

$$E_{\text{data}}(x,y,d) = \min\{E_L(x,y,d), E_R(x,y,d)\} \tag{3}$$

$$E_L(x,y,d) = | I_C(x,y) - I_L(x+d,y) | \tag{4}$$

$$E_R(x,y,d) = | I_C(x,y) - I_R(x-d,y) | \tag{5}$$

where $E_L(x,y,d)$ and $E_R(x,y,d)$ represent the absolute differences between the target and two reference views, respectively. $I(x, y)$ indicates the intensity value at the pixel coordinate (x, y). The minimum matching cost among the two absolute differences is determined by Eq. (3).

$E_{\text{smooth}}(x,y,d)$ denotes the difference between disparity candidates and neighbouring pixels:

$$E_{\text{smooth}}(x,y,d) = \sum_{(x,y)} \rho\{D(x,y) - D(x+1,y)\} + \rho\{D(x,y) - D(x,y+1)\} \tag{6}$$

where ρ indicates the monotonically increasing function of the disparity difference, and $D(x,y)$ represents the disparity at (x,y).

The second step is the error minimization process. The optimal disparity values are determined in this step by comparing matching costs of neighbouring pixels. There are several error minimization techniques, such as Graph-cuts, belief propagation and dynamic programming. In the present paper, we employ Graph-cuts for the optimization process.

The final step is the disparity-to-depth conversion. The disparity map obtained in the second step is transformed to the depth map, represented by an 8-bit greyscale image. The distance information from each camera ranges from 0 to 255, specifying the farthest to the nearest, respectively. The depth value Z of the pixel position (x,y) is transformed into an 8-bit grey value v by:

$$v = \left\lfloor 255 - \frac{255(Z - Z_{\text{near}})}{Z_{\text{far}} - Z_{\text{near}}} + 0.5 \right\rfloor \tag{7}$$

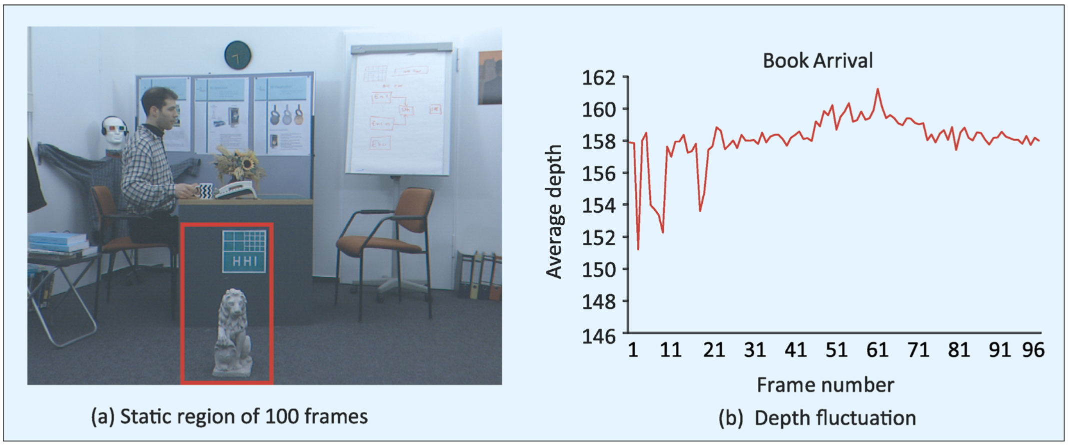
where Z_{far} and Z_{near} represent the farthest and the nearest depth values, respectively.

2.3 Temporally consistent depth estimation

Ideally, for static regions, depth values are identical in each frame if the configuration of the multi-view cameras is fixed. However, since depth estimation is conducted independently on a frame basis, depth values for the static regions fluctuate. The temporal depth inconsistency problem of the video affects the rendering quality of the synthesized views, because flickering artefacts cause viewer discomfort. In addition, performance degradation of temporal prediction occurs in depth coding.

Figure 3 shows average depth values in the static region. Figure 3 (a) shows the static region for 100 frames, and Figure 3 (b) shows the average depth values. As shown in Figure 3 (b), the average depth values fluctuate severely.

There are two major approaches for handling temporal inconsistencies in the depth map: dynamic depth estimation [10-17] and depth video filtering [18-19].



(a) Static region of 100 frames

(b) Depth fluctuation

Fig.3 *Depth fluctuation of static regions for "Book Arrival"*

For dynamic depth map estimation there are several different methods. In Ref. [10], a segment-based dynamic depth estimation algorithm was introduced based on the assumption that a 3D scene is composed of many piecewise planes. Because a certain segment takes some segments from the previous and next frames into account iteratively, it provides reliable depth in the video. Also, because this method is sensitive to depth errors during extraction of initial segments, a temporally consistent depth reconstruction algorithm using enhanced belief propagation was proposed in Ref. [11]. Another approach presented in Ref. [12] describes a spatial-temporal-consistent multi-view depth estimation algorithm. After obtaining the initial disparity map by loopy belief propagation and segmentation based on a plane-fitting process, the spatial-temporal coherence constraint for both the colour and disparity data is defined by a Gaussian distribution. Dynamic depth estimation using a trinocular video system has been recently proposed in Ref. [13]. From the detection process for dynamic regions, spatial-temporal depth optimization is performed adaptively by bundle optimization for static regions and temporal optimization for dynamic regions.

In 3D MPEG video coding, many works have performed temporally consistent depth estimation. A temporal consistency enhancement algorithm was proposed at the 85th and 86th MPEG meetings in 2008 [14-15]. Many experts agreed on the importance of temporal consistency in the depth map, and it was implemented in the depth estimation software [16-17]. However, this algorithm only considers static regions.

Other methods using depth video filtering perform data-adaptive kernel filtering on the depth video. Joint bilateral filtering is the most popular method in this category [18]. The Joint Bilateral Filter (JBF) uses spatial and range weighting functions derived from the coordinate distance and photometric similarity between the target pixel and its neighbours.

In the depth map, suppose there exists a target pixel p and one of its neighbours q. S_p and S_q are depth values at p and q, respectively, and I_p and I_q are the associated colour values at p and q, respectively. The new depth value $\tilde{S}_p$ via JBF is computed by:

$$\tilde{S}_p = \frac{\sum\limits_{q\in\Omega} S_q \cdot f(\| p-q \|) g(\| I_p - I_q \|)}{\sum\limits_{q\in\Omega} f(\| p-q \|) g(\| I_p - I_q \|)} \quad (8)$$

where f and g indicate the spatial and range filters, respectively, and Ω is the local kernel size. If a Gaussian distribution is used to model the weighting function, they are represented by:

$$f(x) = \exp\left(-\frac{x^2}{2\sigma_f{}^2}\right), \quad g(x) = \exp\left(-\frac{x^2}{2\sigma_g{}^2}\right) \quad (9)$$

where σ_f and σ_g are the standard deviations of f

and g, respectively.

Recently, a new method based on 3D-JBF has been proposed to enhance temporal consistency [19]. Specifically, filtering is extended to the temporal domain to reduce temporal fluctuation. Range filters for colour and depth data are applied adaptively based on depth distribution inside the filter kernel. However, the lack of a method for handling temporal motion causes motion blur artefacts.

III. Proposed Method

Figure 4 shows a block diagram of the proposed algorithm. After the matching costs are computed using the current left, centre, and right views, I_L^t, I_C^t, and I_R^t, costs are updated by the temporal weighting function using the reconstructed depth map D'^t_C. The depth map is reconstructed by variable block partitioning, motion estimation and motion compensation. To calculate the motion vector, the current and the previous centre views, I_C^t and I_C^{t-1} are used.

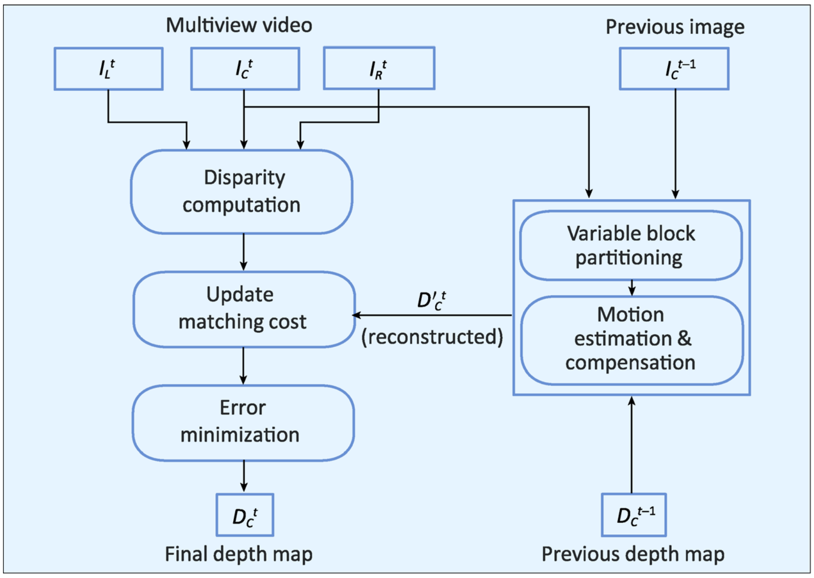


Fig.4 *Block diagram of the proposed algorithm*

As mentioned above, the depth video is temporally inconsistent because the conventional depth estimation algorithm operates independently on a frame basis. Therefore, we modified the matching function using a temporal weighting function. The temporal weighting function using a truncated linear model refers to the reconstructed depth map when estimating the current depth values as described in Eq. (10). The motion estimation process may occasionally fail, resulting in depth errors in the reconstructed depth map. In this case, the temporal weighting function is inefficient. Therefore, we define the temporal weighting function by the truncated linear model to reject outliers. The temporal weighting function $E_{\text{temp}}(x,y,d)$ and the modified matching function are defined by:

$$E_{\text{temp}}(x,y,d)=\min\left\{\,|\,d-D'^t_C(x,y)\,|,L\,\right\} \qquad (10)$$

$$E(x,y,d)=E_{\text{data}}(x,y,d)+\lambda_s E_{\text{smooth}}(x,y,d)+\lambda_t E_{\text{temp}}(x,y,d) \qquad (11)$$

where λ_t indicates the weighting factor for $E_{\text{temp}}(x,y,d)$, L is a constant for outlier rejection and $D_C^t(x,y)$ represents the reconstructed depth value at (x,y).

Since flickering artefacts are most noticeable in the static regions, the proposed method applies a simple block-based object detection method. After the Mean Absolute Difference (MAD) is calculated for each block, the threshold determines whether the block is static. Figure 5 shows the result of moving object detection for the “Book Arrival” sequence.

Then, block-based motion estimation and compensation techniques are used to deal with moving objects. Notice that the block size for motion estimation is smaller in moving object detection for more accurate and reliable motion search. We can reconstruct the current depth map from the previous depth map by detecting moving objects using a larger block size and estimating motion using a smaller block size. Finally, matching costs can be updated by the temporal weighting function using the reconstructed depth map, as described in Eqs. (10-11). Figure 6 shows the reconstructed depth map for the “Book Arrival” sequence.

IV. Experimental Results and Analysis

To evaluate the performance of the proposed

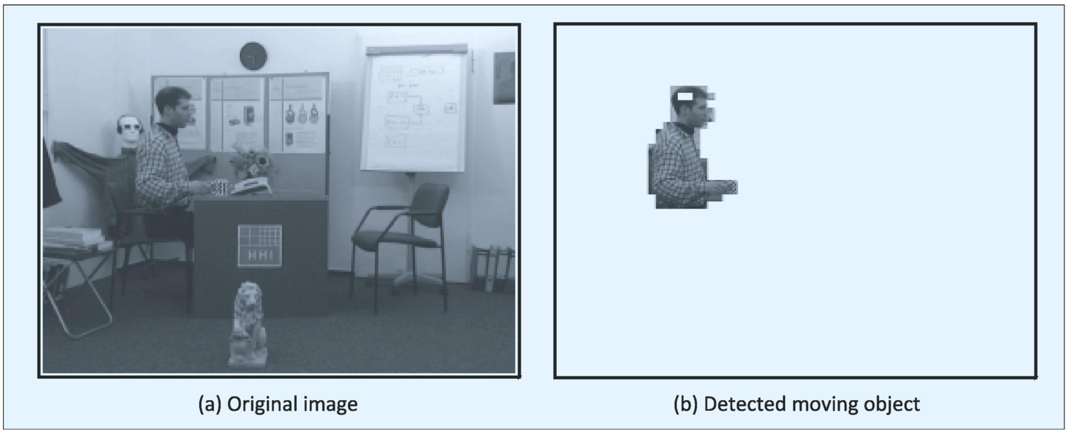


Fig.5 *Moving object detection for "Book Arrival"*

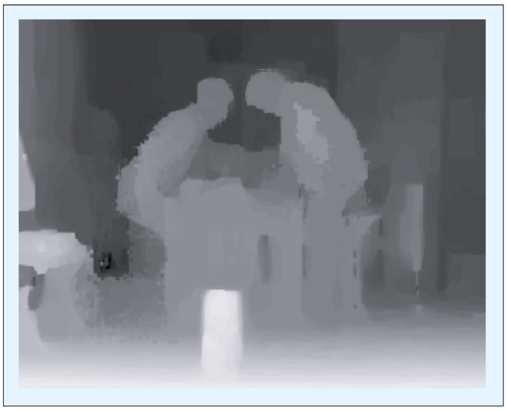


Fig.6 *Reconstructed depth map for "Book Arrival"*

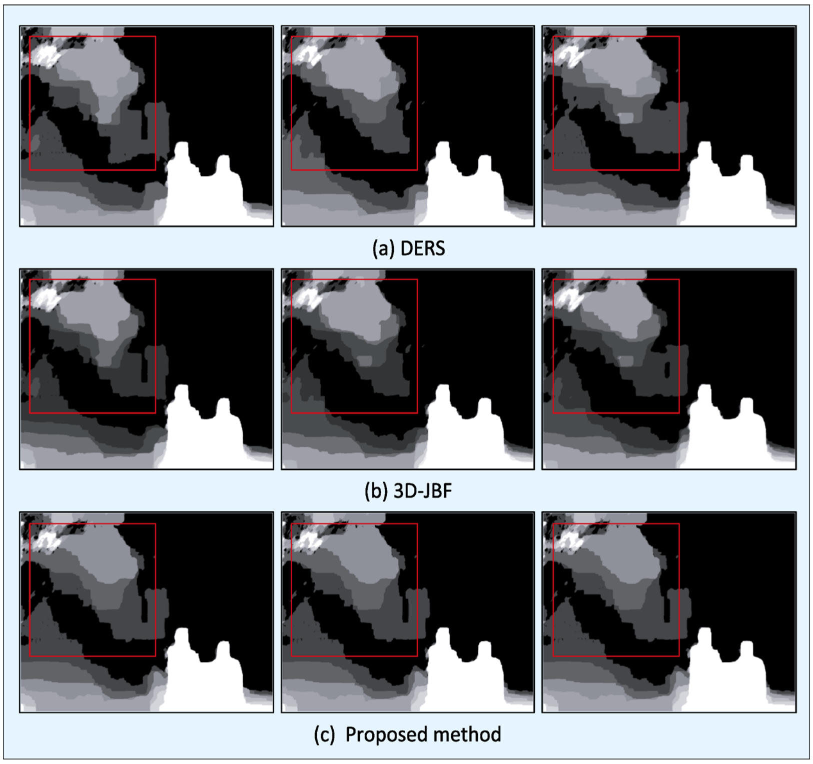


Fig.7 *Depth estimation results for "Lovebird1"*

algorithm, we tested four sequences: "Alt Moabit" and "Book Arrival", provided by the Heinrich-Hertz-Institut (HHI) [20], "Lovebird1" provided by MPEG-Korea Forum [21], and "Newspaper" provided by Gwangju Institute of Science and Technology (GIST) [22]. These sequences are distributed for 3D video testing in MPEG. To obtain depth videos and synthesize virtual views, we used Depth Estimation Reference Software (DERS) 5.0 and View Synthesis Reference Software (VSRS) 3.5 [16]. We compared the proposed method with DERS and 3D-JBF [19]. The number of frames was 100, and the block sizes were 32 × 32 for the static region and 8 × 8 for the moving object. The search range for motion estimation was from −16 to +16. The weighting factors, λ_s and λ_t, and the threshold for moving object detection were sequence-dependent. Through several experiments, we have determined the block sizes and threshold values for moving object detection that obtain reliable rendering quality.

4.1 Depth estimation and view synthesis

Figures 7 and 8 show the depth sequences for "Lovebird1" and "Newspaper", respectively. From Figure 7 (a) and (b) and Figure 8 (a) and (b), we noticed that the depth sequences have inconsistent depths in the static regions, whereas the depth sequences in Figures 7 (c) and 8 (c) are temporally consistent.

To verify the rendering quality of the proposed method quantitatively, we checked the amount of depth fluctuation by calculating the average depth values in static regions; smaller depth fluctuation guarantees higher rendering quality. Figure 9 shows static regions of 100 frames for each sequence.

Figure 10 shows the average depth values of the static regions, as indicated in Figure 9. The dotted line, dashed line and solid line represent the results of DERS, 3D-JBF and the proposed method, respectively. As shown in Figure 10, the average depth values fluctuated in the conventional work, whereas the proposed method reduced depth fluctuation.

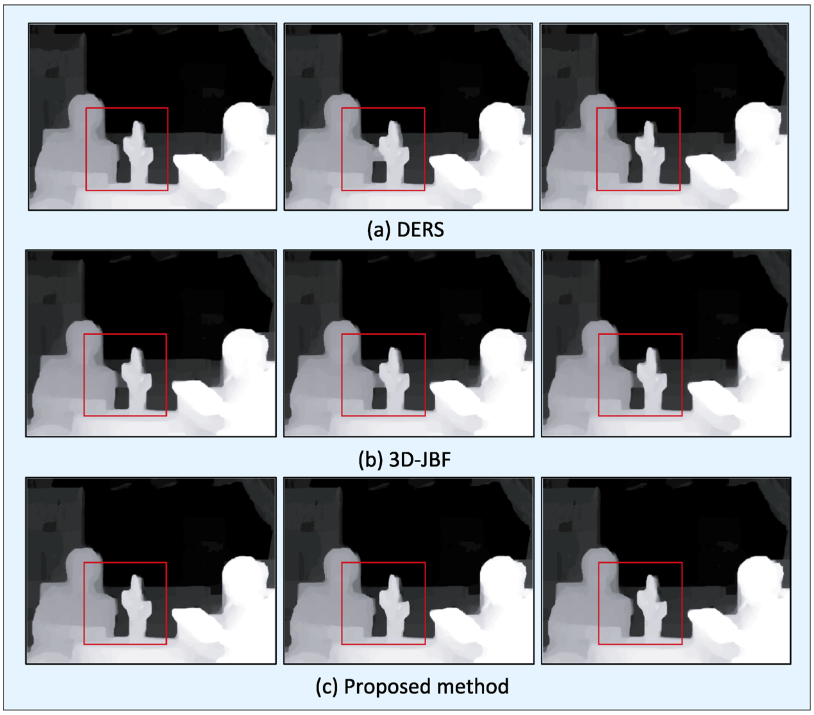
4.2 Depth video coding

To evaluate the performance of the proposed method for depth video coding, we encoded several different depth videos using the H.264/AVC reference software, JM 14.0 [23]. The number of frames for each sequence was 100, and the coding structure was IPPP...P. The Quantization Parameters (QP) were determined independently for each sequence to indicate noticeable quality drops.

Figure 11 and Table I show the depth video coding results. The Picture Signal-to-Noise Ratio (PSNR) was calculated from the original view and the synthesized view. From Figure 11 and Table I, we can see that the rendering quality of the proposed algorithm was preserved, while the bitrates were reduced by 38.19% on average.

We examined the mode information change of the depth video coding with QP 22. As shown in Figure 12, intra modes were changed to inter modes, and larger blocks were selected. From these results, we can infer that the temporal prediction accuracy of H.264/AVC was increased because the proposed method improved the temporal consistency of the depth video and reduced depth fluctuation. As a result, bitrates were much less than when DERS was used.

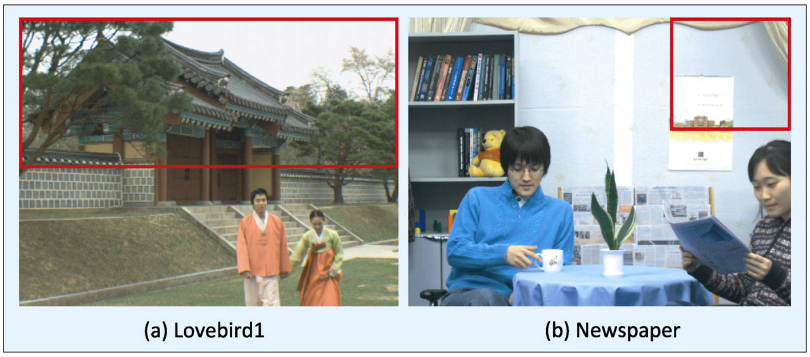
To analyse further the relationship between depth consistency and rendering quality, we encoded the colour and depth videos of the "Newspaper" sequence using various QPs. Then, we generated synthesized images with every possible combination of the encoded colour and depth videos. Figure 13 and Table II show the experimental results of the video coding


(a) DERS

(b) 3D-JBF

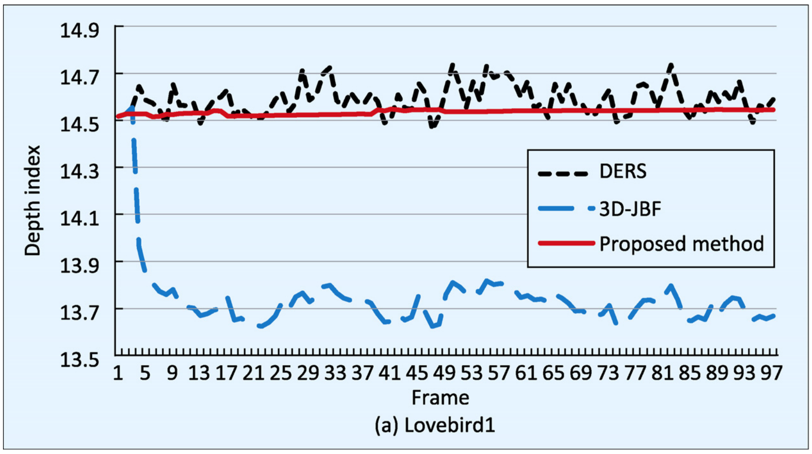
(c) Proposed method

Fig.8 *Depth estimation results for "Newspaper"*

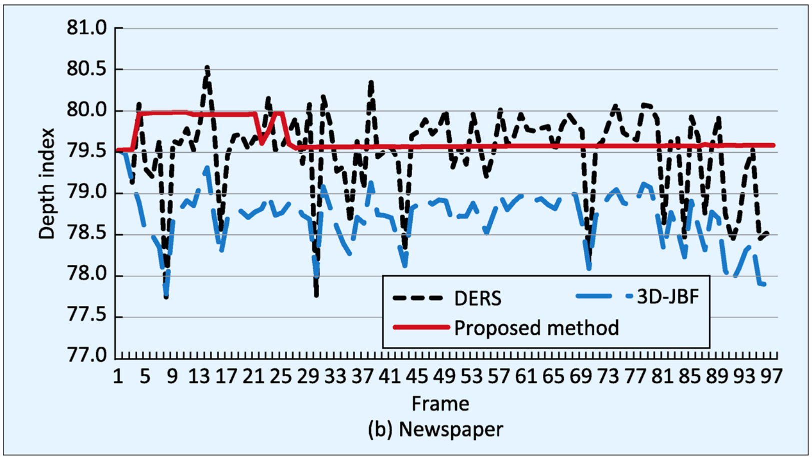

(a) Lovebird1

(b) Newspaper

Fig.9 *Static regions of* 100 *frames*



(a) Lovebird1



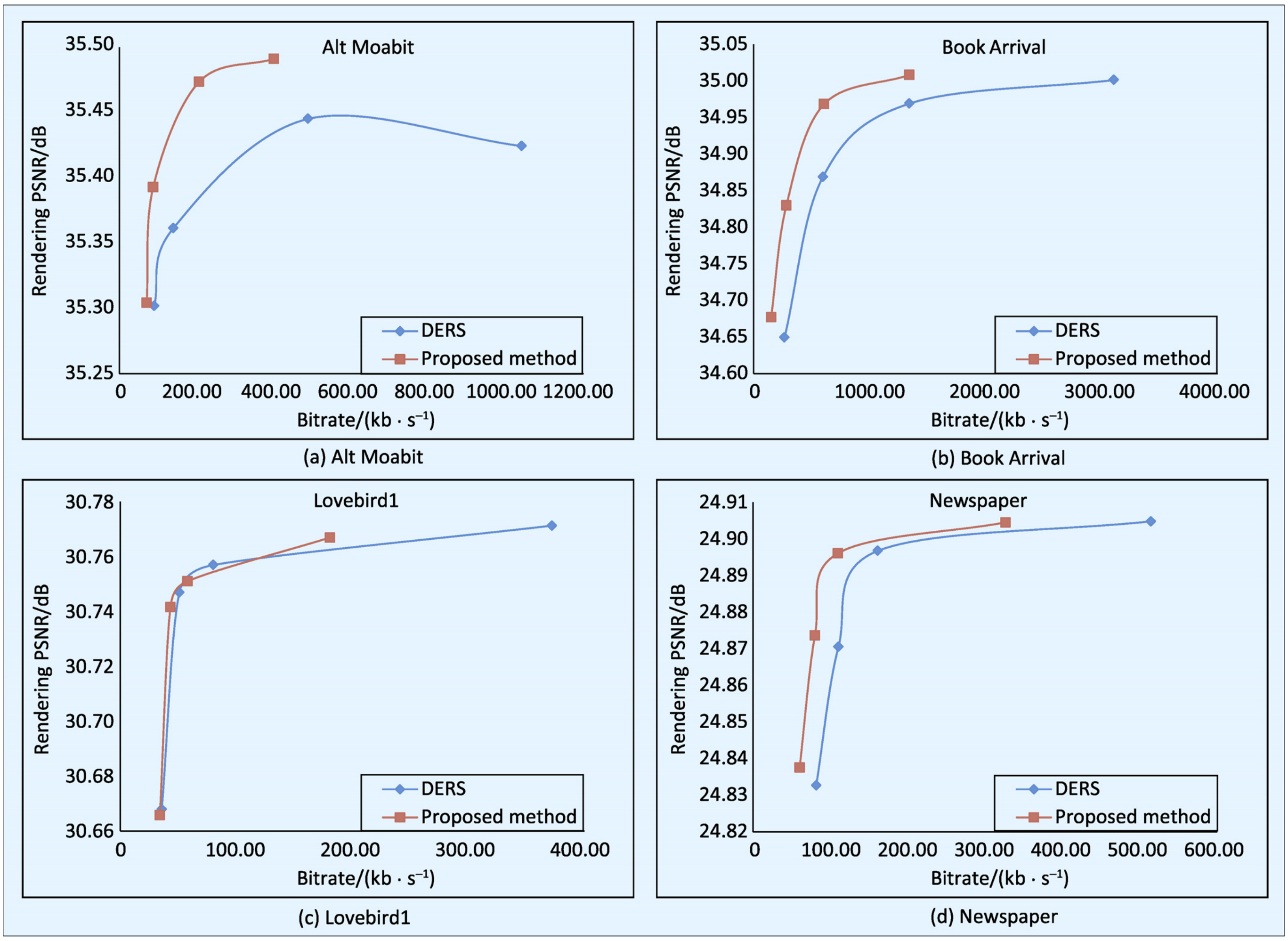
(b) Newspaper

Fig.10 *Average depth variation*

and view synthesis. From these results, we notice that the rendering PSNR does not drop steeply as the QP of the depth video increases. In other words, the rendering PSNR is rarely sensitive to depth quality. Therefore, we infer that the proposed method definitely improves depth consistency and temporal prediction of depth video coding while preserving rendering quality.

V. Conclusion

In the present paper, we proposed a temporally consistent multi-view depth estimation algorithm. The proposed method exploited the temporal weighting function that takes the motion-compensated depth map into account. We used moving object detection and motion estimation with variable block sizes to obtain a motion-compensated depth map. The experimental results demonstrated that the proposed method generated temporally consistent depth video and reduced flickering artefacts. Consequently, the proposed method preserved good rendering quality and reduced the number of coding bits by about 38% on average.



(a) Alt Moabit

(b) Book Arrival

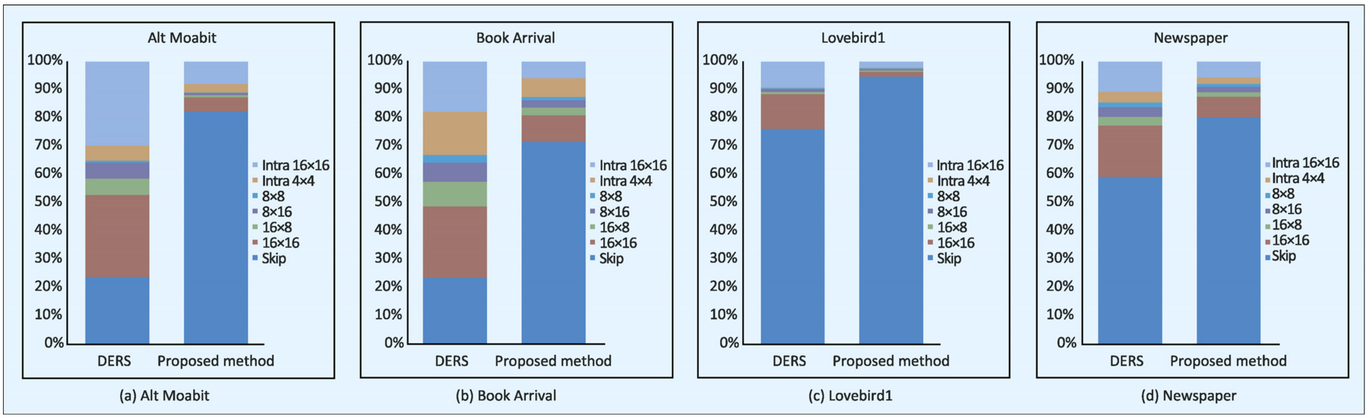
(c) Lovebird1

(d) Newspaper

Fig.11 *Rate-rendering distortion curves for depth video coding*

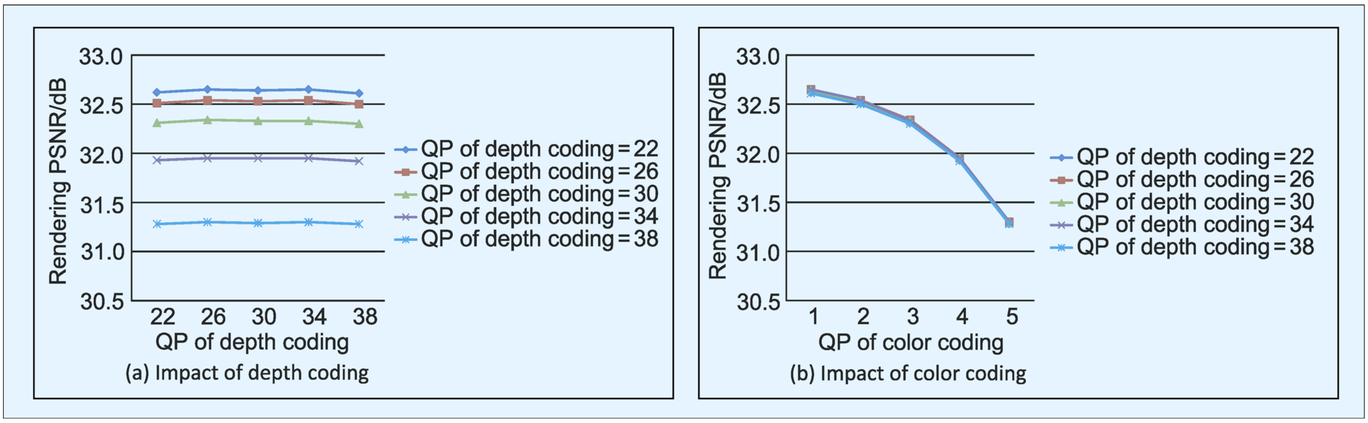
Table I *Results of depth video coding*

Sequence	QP	DERS		Proposed method		Bitrate reduction/%	ΔPSNR/dB
		Average bitrate/(kb·s^{-1})	Average rendering PSNR/dB	Average bitrate/(kb·s^{-1})	Average rendering PSNR/dB		
Alt Moabit (Synthesized view: 8, 9) (Depth video view: 7, 10)	25	1 047.98	35.42	401.55	35.49	61.68	0.07
	31	490.33	35.44	206.01	35.47	57.99	0.03
	40	138.67	35.36	86.01	35.39	37.98	0.03
	43	88.56	35.30	69.16	35.30	21.90	0.00
Book Arrival (Synthesized view: 8, 9) (Depth video view: 7, 10)	22	3 128.92	35.00	1 350.29	35.01	56.84	0.01
	28	1 350.34	34.97	608.28	34.97	54.95	0.00
	34	599.01	34.87	280.66	34.83	53.15	−0.04
	40	264.87	34.65	147.87	34.68	44.17	0.03
Lovebird1 (Synthesized view: 6, 7) (Depth video view: 5, 8)	22	375.67	30.77	182.57	30.77	51.40	0.00
	31	81.00	30.76	58.64	30.75	27.60	−0.01
	34	51.47	30.75	43.71	30.74	15.08	−0.01
	37	36.22	30.67	34.54	30.67	4.64	0.00
Newspaper (Synthesized view: 4, 5) (Depth video view: 3, 6)	28	518.03	24.90	328.18	24.90	36.65	0.00
	37	161.42	24.90	109.22	24.90	32.34	0.00
	40	110.12	24.87	79.37	24.87	27.92	0.00
	43	81.04	24.83	59.42	24.84	26.67	0.00
Average						**38.19**	**0.01**



(a) Alt Moabit (b) Book Arrival (c) Lovebird1 (d) Newspaper

Fig.12 *Mode information change of depth video coding*



(a) Impact of depth coding (b) Impact of color coding

Fig.13 *Relationship between rendering PSNR and depth quality*

Table II *Results of video coding and view synthesis*

Rendering PSNR/dB		Depth coding				
		QP=22	QP=26	QP=30	QP=34	QP=38
Color coding	QP=22	32.62	32.65	32.64	32.65	32.61
	QP=26	32.51	32.54	32.53	32.54	32.50
	QP=30	32.31	32.34	32.33	32.33	32.30
	QP=34	31.93	31.95	31.95	31.95	31.92
	QP=38	31.28	31.30	31.29	31.30	31.28

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea Grant funded by the Korea Ministry of Science and Technology under Grant No. 2012-0009228.

References

[1] BARFIELD W, WEGHORST S. The Sense of Presence within Virtual Environments: A Conceptual Framework[J]. Human Computer Interaction: Software and Hardware Interfaces, 1993, 2: 699-704.

[2] FREEMAN J, AVONS S E. Focus Group Exploration of Presence through Advanced Broadcast Services[J]. SPIE, Human Vision and Electronic Imaging, 2000, 3953: 3959-3976.

[3] SOMLIC A, MUELLER K, MERKLE P, *et al.* 3D Video and Free Viewpoint Video - Technologies, Applications and MPEG Standards[C]// Proceedings of 2006 IEEE International Conference on Multimedia and Expo: July 9-12, 2006. Toronto, Canada. 2006: 2161-2164.

[4] REDERT A, BEECK M O, FEHN C, *et al.* ATTEST: Advanced Three-Dimensional Television System Technologies[C]// Proceedings of the 1st International Symposium on 3D Data Processing Visualization and Transmission: June 19-21, 2002. Padova, Italy, 2002: 24-36.

[5] RIVA G, DAVIDE F, IJSSELSTEIJN W A. Being There: Concepts, Effects and Measurement of User Presence in Synthetic Environments[M]. Amsterdam, the Netherlands: IOS Press, 2003.

[6] SCHARSTEIN D, SZELISKI R ZABIH R. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms[C]// Proceedings of IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001): December 9-10, 2001. Kauai, Hawaii, USA, 2001: 131-140.

[7] ZITNICK C L, KANG S B, UYTTENDAELE M, *et al.* High-Quality Video View Interpolation Using a Layered Representation[C]// Proceedings of the 31st International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH): August 8-12, 2004. Los Angeles, CA, USA, 2004: 600-608.

[8] LEE E K, HO Y S. Generation of High-Quality Depth Maps Using Hybrid Camera System for 3-D Video[J]. Journal of Visual Communication and Image Representation, 2011, 22(1): 73-84.

[9] LEE C, SONG H, CHOI B, *et al.* 3D Scene Capturing Using Joint Stereo Camera with a Time-of-Flight Camera for 3D Displays[J]. IEEE Transactions on Consumer Electronics, 2011, 57(3): 1370-1379.

[10] TAO Hai, SAWHNEU H S, RAKESH K. Dynamic Depth Recovery from Multiple Synchronized Video Streams[C]// Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001): International Conference on Image Processing: December 8-14, 2001. Kauai, Hawaii, USA, 2001: 118-124.

[11] LARSEN E S, MORDOHAI P, POLLEFEYS M, *et al.* Temporally Consistent Reconstruction from Multiple Video Streams Using Enhanced Belief Propagation[C]// Proceedings of IEEE 11th International Conference on Computer Vision (ICCV 2007): October 14-21, 2007. Rio de Janerio, Brazil, 2007: 1-8.

[12] YANG Mingjin, CAO Xue, DAI Qionghai. Multiview Video Depth Estimation with Spatial-Temporal Consistency[C]// Proceedings of British Machine Vision Conference (BMVC 2010): August 31-September 3, 2010. Aberystwyth, UK, 2010: 1-11.

[13] YANG Wenzhou, ZHANG Guofeng, BAO Hujun, *et al.* Consistent Depth Maps Recovery from a Trinocular Video Sequence[C]// Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR): June 16-21, 2012. Providence, RI, USA, 2012: 1466-1473.

[14] LEE S B, HO Y S. Enhancement of Temporal Consistency for Multi-View Depth Map Estimation[R]. ISO/IEC JTC1/SC29/WG11, M15594, 2008.

[15] LEE S B, HO Y S. Experiment on Temporal Enhancement for Depth Estimation[R]. ISO/IEC JTC1/SC29/WG11, M15852, 2008.

[16] TANIMOTO M, FUJII T, SUZUKI K. Reference Software of Depth Estimation and View Synthesis for FTV/3DV[R]. ISO/IEC JTC1/SC29/WG11, M15836, 2008.

[17] LEE S B, LEE C, HO Y S. Experimental Results on Improved Temporal Consistency Enhancement[R]. ISO/IEC JTC1/SC29/WG11, M16063, 2009.

[18] KOPF J, COHEN M F, LINCKINSKI D, *et al*. Joint Bilateral Upsampling[C]// Proceedings of the 34th International Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH): August 5-9, 2007. San Diego, CA, USA, 2007: 96-100.

[19] CHOI J, MIN Dongbo, SOHN K. 2D-Plus-Depth Based Resolution and Frame-Rate Up-Conversion Technique for Depth Video[J]. IEEE Transactions on Consumer Electronics, 2010, 56(4): 2489-2497.

[20] FELDMANN I, MUELLER M, ZILLY F, *et al*. HHI Test Material for 3D Video[R]. ISO/IEC JTC1/SC29/WG11, M15413, 2008.

[21] UM G M, BANG G, HUR N, *et al*. Contribution for 3D Video Test Material of Outdoor Scene[R]. ISO/IEC JTC1/SC29/WG11, M15371, 2008.

[22] HO Y S, LEE E K, LEE C. Multi-View Video Test Sequence and Camera Parameters[R]. ISO/IEC JTC1/SC29/WG11, M15419, 2008.

[23] http://iphome/hhi.de/shehring/tml/download/old_jm/jm14.0.zip, Joint Video Team, Reference Software Version 14.0.

Biographies

Sang-Beom Lee, received his B.S. degree in electric and electronic engineering from Kyungpook National University (KNU), Korea in 2004 and M.S. degree in information and communication engineering at the Gwangju Institute of Science and Technology (GIST), Korea in 2006. He is currently working towards his Ph.D. degree in the Department of Information and Communications at GIST, Korea. His research interests include digital image processing, 3D television, depth estimation, 3D video coding, and realistic broadcasting. Email: sblee@gist.ac.kr

Yo-Sung Ho, received both B.S. and M.S. degrees in electronic engineering from Seoul National University (SNU), Korea in 1981 and 1983, respectively, and 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, New York. In 1993, he rejoined the technical staff of ETRI. Since 1995, he has been with the Gwangju Institute of Science and Technology (GIST), where he is currently a Professor in the School of Information and Communications. His research interests include digital image and video coding, advanced coding techniques, 3D television, and realistic broadcasting. *The corresponding author. Email: hoyo@gist.ac.kr