

특집논문 (Special Paper)

방송공학회논문지 제20권 제4호, 2015년 7월 (JBE Vol. 20, No. 4, July 2015)

<http://dx.doi.org/10.5909/JBE.2015.20.4.545>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

3D-HEVC를 위한 인터-컴포넌트 부호화 방법

박민우^{a)}, 이진영^{a)}, 김찬열^{a)†}

Overview of Inter-Component Coding in 3D-HEVC

Min Woo Park^{a)}, Jin Young Lee^{a)}, and Chanyul Kim^{a)†}

요약

차세대 3차원 디스플레이 및 서비스를 지원하기 위한 HEVC 기반 3차원 비디오 코딩 표준(3D-HEVC)이 최근 완료되었다. 3D-HEVC는 소수의 텍스처 영상(Texture image)과 깊이 영상(Depth map image)으로 구성된 Multi-view plus depth (MVD) 포맷을 효율적으로 처리하기 위한 표준으로써 H.264/AVC와 HEVC에서 사용하는 단일 계층 부호화 방법과 더불어 텍스처 영상들간, 깊이 영상들간, 텍스처 영상과 깊이 영상들간의 예측을 수행하는 인터-컴포넌트 부호화 기술을 추가적으로 사용한다. 본 논문에서는 3D-HEVC 표준의 일반적인 코딩 구조, 3D-HEVC 기술의 기반이 되는 인터-컴포넌트 부호화 기술 및 인터-컴포넌트 부호화 효율에 중요한 영향을 미치는 시차 벡터(Disparity vector) 유도 기술에 대해 상세히 소개한다. 또한 본 논문에서는 3D-HEVC의 부호화 효율을 검증하기 위해 각 시점을 HEVC로 부호화한 방법과 단순 다시점 확장 표준인 MV-HEVC와의 성능평가를 수행한다.

Abstract

A HEVC-compatible 3D video coding method (3D-HEVC) has been recently developed as an extension of the high efficiency video coding (HEVC) standard. In order to efficiently deal with the multi-view video plus depth (MVD) format, 3D-HEVC exploits an inter-component prediction which allows the prediction between texture and depth map images in addition to a temporal prediction used in the conventional single layer video coding such as H.264/AVC and HEVC. The performance of the inter-component prediction is normally affected by the accuracy of the disparity vector, and thus it is important to have an accurate disparity vector used for the inter-component prediction. This paper, therefore, introduces a disparity derivation method and inter-component algorithms using the disparity vector for the efficient 3D video coding. Simulation results show that the 3D-HEVC provides higher coding performance compared with the simulcast approach using HEVC and the simple multi-view extension (MH-HEVC).

Keyword : 3D video coding, 3D-HEVC, disparity vector derivation, inter-component prediction

a) 삼성전자 DMC연구소 Advanced Media Lab.(Advanced Media Lab., DMC R&D Center, Samsung Electronics)

† Corresponding Author : 김찬열(Chanyul Kim)

E-mail: dionism@gmail.com

Tel: +82-31-279-9047

ORCID: <http://orcid.org/0000-0002-2337-5177>

Manuscript received May 19, 2015; revised June 24, 2015; accepted June 24, 2015.

1. Introduction

This paper introduces a 3D video coding method (3D-HEVC) as an extension of the high efficiency video coding (HEVC) standard, which has been recently developed in the ITU-T/ISO/IEC Joint Collaborative Team on

3D Video Coding Extension Development (JCT-3V) standardization group [1]. Along with the advances in 3D video displays, such as a glasses-free 3D television, a free-view point television (FTV) [2], and so on, these displays need to have multiple views (more than two views) to provide more realistic 3D experience to users. In order to represent a glasses-free 3D video content on these kinds of displays, a relatively large number of views need to be sent and displayed, because it needs to display a stereo view pair in all the positions within the possible viewing angles. However, in the real 3D application such as a 3D video broadcasting and a 3D video player it is far more difficult to handle a large number of views because of the limitation of bandwidth or capacity of the digital media.

In order to deal with the multiple views efficiently, a multi-view video plus depth (MVD) format as a new 3D video representation format was introduced. MVD consists of only a small number (two or three) of texture images and their associated depth map images, and corresponding camera parameters [3]. The MVD format allows a client to generate intermediate texture views between received and reconstructed texture images by using their associated depth map images based on the view synthesis algorithm such as the depth-image-based rendering (DIBR) technique [4]. As the new 3D representation format was introduced and the HEVC standard, which has double the compression ratio compared to the Advanced Video Coding (H.264/AVC) standard [5] at the same visual quality level, was finalized in January 2013 [6], the JCT-3V standardization group started to develop the HEVC-compatible 3D video coding standard as the 3D extension of HEVC for efficient compression of the MVD format and finalized it in February 2015.

The multi-view video coding standard such as the Multi-view Video Coding (MVC) standard [5] uses an inter-view prediction, which only allows the prediction across multiple texture view images in the same time instance,

without block-level changes. However, 3D-HEVC exploits an inter-component prediction which additionally allows the prediction across texture and depth map images in addition to the inter-view prediction. This paper introduces a disparity derivation method and inter-component coding tools using the disparity in 3D-HEVC. The inter-component algorithms consist of depth-based disparity vector derivation, inter-view motion information prediction, view synthesis prediction, illumination compensation, residual prediction, and depth-based block partitioning. The results demonstrate that 3D-HEVC improves the coding performance by exploiting the inter-component coding tools.

The remaining paper is organized as follows. Section II presents the general coding structure of 3D-HEVC. Section III explains the depth-based disparity vector derivation method. In section IV, the inter-component coding tools are explained in detail. Finally, section V shows a simulation results and section VI concludes the paper, respectively.

II. Coding Structure and Algorithms

Figure 1 illustrates an example of a general coding structure of 3D-HEVC, which normally has three texture images with their associated depth map images as shown in Fig. 1. The texture image of view 0 in Fig. 1, which is a base view, is always encoded with the HEVC encoder to have a backward compatibility with HEVC. The dependent view (view 1 and view 2 in Fig. 1) can use the inter-component predictions between texture and depth map images in the same time instance, that is, the same access unit (AU), as shown in Fig. 1.

After encoding of all texture and depth map images, the encoded data is integrated by a multiplexer to a 3D-HEVC bitstream [7]. The 3D-HEVC bitstream is designed to simultaneously support various view-points such as a 2D

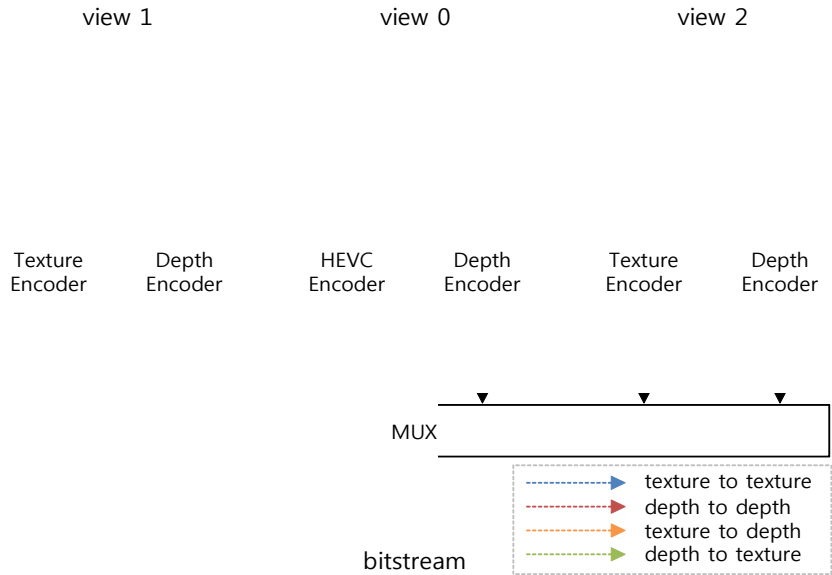


그림 1. 3D-HEVC의 일반적인 부호화 구조
 Fig. 1. General coding structure of 3D-HEVC

video display, a stereoscopic video display and a glasses-free 3D video display by the sub-stream extraction process [6].

For instance, for displaying a 3D video content on a conventional 2D video display, only the texture image of view 0 can be extracted from the 3D-HEVC bitstream and then the extracted bitstream can be decoded using the HEVC decoder. For a stereoscopic video display, two texture views of either view 0 and view 1, or view 0 and view 2 can be extracted from the 3D-HEVC bitstream and then the extracted bitstream can be decoded using the 3D-HEVC

decoder. For a glasses-free 3D display, the 3D-HEVC bitstream can be decoded all the texture and depth map images using the 3D-HEVC decoder and then N intermediate views can be additionally generated by using reconstructed texture and depth map images.

III. Disparity Vector Derivation

In order to access the coded information such as motion pa-

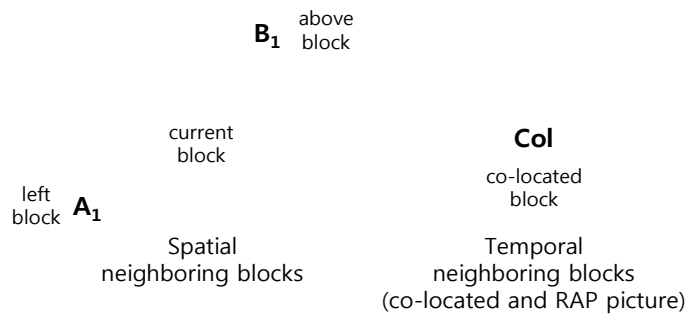


그림 2. 시차 벡터 유도를 위한 주변 블록들
 Fig. 2. Neighboring blocks for disparity vector derivation

rameters and reconstructed samples of the texture and the depth map images in the reference views, the disparity vector needs to be derived for both the texture and the depth map coding. Moreover, the performance of the inter-component prediction is normally affected by the accuracy of the disparity vector, and thus it is important to derive the accurate disparity vector.

The disparity vector derivation for the texture coding consists of two steps; the first step is the derivation from neighboring blocks and the second step is the refinement of the derived disparity vector [8]-[10].

The disparity vector is firstly derived from the temporal and spatial neighboring blocks as shown in Fig. 2. The temporal neighboring blocks are the co-located blocks in the co-located picture defined in the slice header and the random access point (RAP) picture which only has the blocks which performed the inter-view prediction and the spatial neighboring blocks are the left and the above blocks of the current block [11]-[12]. The detailed operation of the first step is illustrated as follows.

- Step 1. A variable availableDV, which indicates whether the disparity vector is found or not, is set to zero.
- Step 2. For two temporal and two spatial neighboring blocks, if availableDV is equal to zero, the disparity vector is found from the neighboring blocks which performed the inter-view prediction, and once the disparity vector is found availableDV is set to one.
- Step 3. For two spatial neighboring blocks, if available DV is equal to zero, the disparity vector from the neighboring blocks which performed the temporal prediction and stored the disparity vector, and once the disparity vector is found availableDV is set to one.
- Step 4. If availableDV is equal to zero, the disparity vector is set to (0, 0) by default [13].

After obtaining the disparity vector from neighboring blocks, the derived disparity vector is refined by the corresponding depth block. Since the depth map image of the reference view is already available, more accurate disparity vector can be derived from the corresponding depth block in the reference view. Fig. 3 shows how to derive the corresponding depth block. The corresponding depth block is detected by the disparity vector derived from neighboring

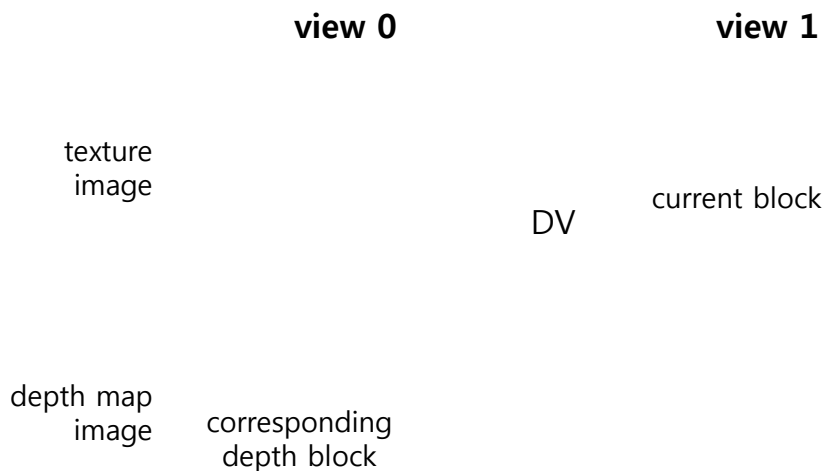


그림 3. 현재 블록에 대응하는 깊이 블록 유도 방법
Fig. 3. Derivation of corresponding depth block

blocks (DV in Fig. 3). The representative depth value among the four corner depth samples in the corresponding depth block is chosen and then is converted into the disparity vector by using the camera parameters of the current view. And the vertical component of the refined disparity vector is set to zero because the depth value only considers a horizontal shifting [14]. If the corresponding depth block is located outside or on the boundary of the depth map picture, the sample position of the corresponding block located outside of the picture is clipped to the boundary position of the picture [15].

The disparity vector is also needed for the depth map coding. However, the accuracy of the disparity vector for the depth map block coding is less important than the texture block coding. Therefore, the disparity vector is derived in slice level. The constant depth value 128, which means the middle value of the depth value range in 8 bit-depth image, is converted into the disparity vector by using corresponding camera parameters. The derived disparity vector is used for the whole blocks in a slice [16].

IV. Algorithms Using Inter-Component Prediction

The disparity vector described in the previous section is

used for the coding algorithms which exploit the inter-component prediction. In this section, several inter-component coding tools using the disparity vector are discussed.

1. Inter-view motion information prediction

The motion information of the corresponding block in the reference view pointed by the disparity vector can be used as the motion information of the current block, because the motion information between the current block and the corresponding block is likely to be very similar if they have the same object. The inter-view motion information prediction (IVMP) mode is added as an additional merge candidate into the merge candidate list.

For the texture block coding, the motion information derivation can be conducted in sub-block level from the corresponding block as shown in Fig.4. The current block is divided into 8x8 sub-blocks and, the motion information is derived for each sub-block and then the motion compensation is performed for each sub-block using the derived motion information [17]. The motion information of the bottom-right block of the corresponding block can be additionally used as a merge candidate.

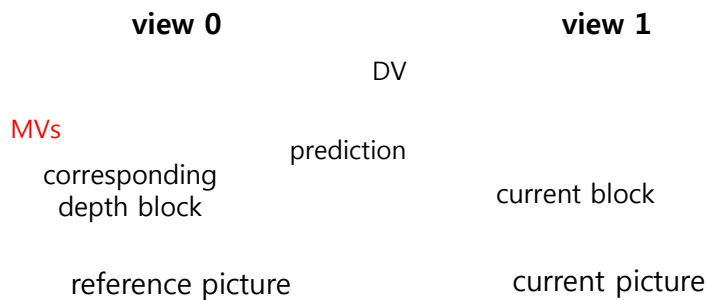


그림 4. 서브블록 단위 시점간 움직임 정보 예측 방법의 예
 Fig. 4. Example of IVMP in sub-block level

Since the IVMC modes always performs the temporal prediction, when the tool which only performs the inter-view prediction such as the illumination compensation is used, then the IVMC mode is not inserted into merge candidate list [18].

2. View synthesis prediction

The view synthesis prediction (VSP) is the sub-block level inter-view prediction tool using the depth information. Fig. 5 shows an example of the VSP process. The current block is firstly divided into the sub-blocks. The sub-block size is decided to either 8x4 or 4x8 block by comparing the gradient of four corner depth samples in the corresponding depth block of the reference view pointed by the disparity vector from neighboring blocks. And then the disparity vector for each sub-block is derived through the corresponding depth sub-block in the same manner with the second step of the disparity derivation process. After obtaining the disparity vector for each sub-block, the disparity compensation prediction for each sub-block is conducted by using the disparity vector [19]-[20].

VSP is used as an additional merge candidate in the

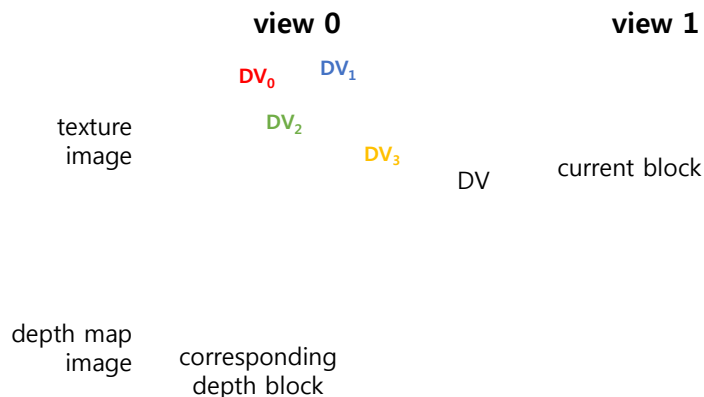


그림 5. 시점 합성 예측 방법의 예
Fig. 5. Example of VSP

merge candidate list. There are two types of VSP candidate; one is the default VSP candidate and the other is the derived VSP candidate from spatial neighboring blocks. If a spatial neighboring merge candidate is coded with VSP, then the corresponding candidate is set to the VSP candidate as the derived VSP candidate. Otherwise, the default VSP candidate is set to the VSP candidate.

3. Illumination compensation

The illumination compensation (IC) is used to compensate for the illumination mismatch between the current tex

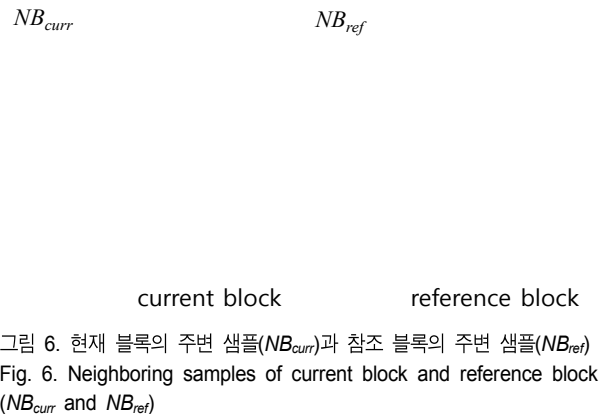


그림 6. 현재 블록의 주변 샘플(NB_{curr})과 참조 블록의 주변 샘플(NB_{ref})
Fig. 6. Neighboring samples of current block and reference block (NB_{curr} and NB_{ref})

ture block and the reference texture block in the reference view picture [21]-[22]. IC is performed by the linear regression model using the neighboring samples (NB_{curr} and NB_{ref}) of the current and reference blocks as shown in Fig. 6 expressed as following equation.

$$Pred'(x, y) = a \times Pred(x, y) + b \quad (1)$$

where (x, y) represents a sample position in the reference block, $Pred(x, y)$ and $Pred'(x, y)$ represent a sample value of the reference block in the reference view and an illumination compensated sample value of the reference block, respectively, a indicates a scale factor and b indicates an offset of the linear regression model.

The scaling factor (a) and the offset (b) of the linear regression model can be calculated as follows:

$$a = \frac{N \cdot \sum_{i=0}^{N-1} NB_{curr}(i) \cdot NB_{ref}(i) - \sum_{i=0}^{N-1} NB_{curr}(i) \cdot \sum_{i=0}^{N-1} NB_{ref}(i) + \lambda}{N \cdot \sum_{i=0}^{N-1} NB_{ref}(i) \cdot NB_{ref}(i) - \left(\sum_{i=0}^{N-1} NB_{ref}(i) \right)^2 + \lambda} \quad (2)$$

$$b = \frac{\sum_{i=0}^{N-1} NB_{curr}(i) - a \cdot \sum_{i=0}^{N-1} NB_{ref}(i)}{N} \quad (3)$$

where N represents the number of available neighboring samples, $NB_{curr}(i)$ and $NB_{ref}(i)$ mean a neighboring sample of the current block and the reference block, respectively. λ means the error term, which considers outliers in neighboring samples.

IC is applied only when the partition mode of the current block is PART_2Nx2N mode. In addition, IC does not perform together with the residual prediction to reduce the complexity in terms of the number of operations and the required external memory bandwidth. In order to further optimize IC and the residual prediction, when the residual prediction is enabled, IC is disabled [23].

4. Residual prediction

The residual prediction (RP) predicts the residual data of the current block from the residual data of the corresponding block in the reference view pointed by the disparity vector from the neighboring blocks.

The residual data of the corresponding block is not stored in the memory, but is derived at the time of decoding of the current block in an on-the-fly manner. The corresponding block

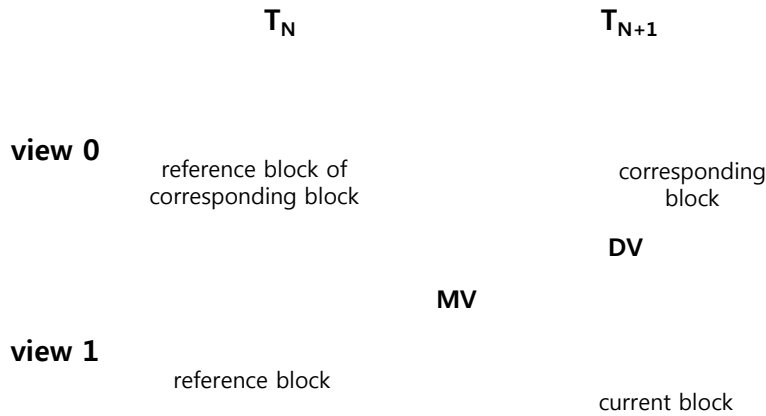


그림 7. 시간 방향 예측 시 잔차 예측 방법
 Fig. 7. RP for temporal prediction

is identified by the disparity vector for the current block and the residual data of the corresponding block is calculated by performing the motion compensation using the same motion information of the current block, and then the derived residual data is added to the reference block data. RP for the temporal prediction is illustrated in Fig. 7 [24].

Additionally, RP is extended to the inter-view prediction cases. When the current block performs the inter-view prediction, the residual data of the corresponding block pointed by the motion vector which is derived the similar method to the disparity vector derivation can be generated by conducting the disparity compensation using the disparity vector for the current block, and then the derived residual data is also added to the reference block data. RP for the inter-view prediction is illustrated in Fig. 8.

In order to perform RP, total three blocks are needed to fetch from the external memory; the reference block, the corresponding block and the reference block of corresponding block for each reference list (list 0 and list 1). Therefore, the bi-linear interpolation filter is used when obtaining the reference block, the corresponding block and

the reference block of the corresponding block to reduce the memory bandwidth requirement in the worst case. In order to further reduce the memory bandwidth requirement, when the current block is 8x8 block size, RP is disabled for chroma component such that the worst case memory bandwidth of 3D-HEVC is less than that of HEVC.

5. Depth-based block partitioning

The depth-based block partitioning (DBBP) mode enables an arbitrary block partitioning prediction using the corresponding depth block pointed by the disparity vector [25]. In order to obtain the arbitrary block partition, a binary segmentation mask is generated based on the corresponding depth block in the reference view. A threshold is firstly derived by averaging four corner samples in the corresponding depth block [26], and then the binary segmentation mask can be generated by using the derived threshold. The depth samples that are greater than the threshold are set to one, which means the foreground region. The other samples are set to zero, which means the background region. Fig. 9 shows an example of generating

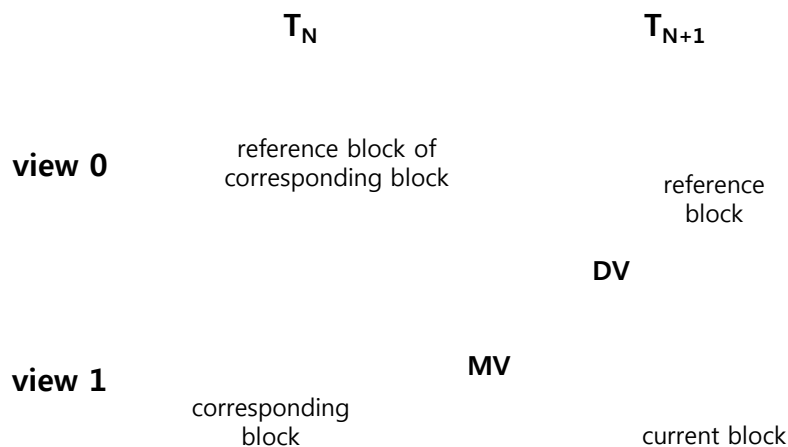


그림 8. 시점 방향 예측 시 잔차 예측 방법
Fig. 8. RP for inter-view prediction

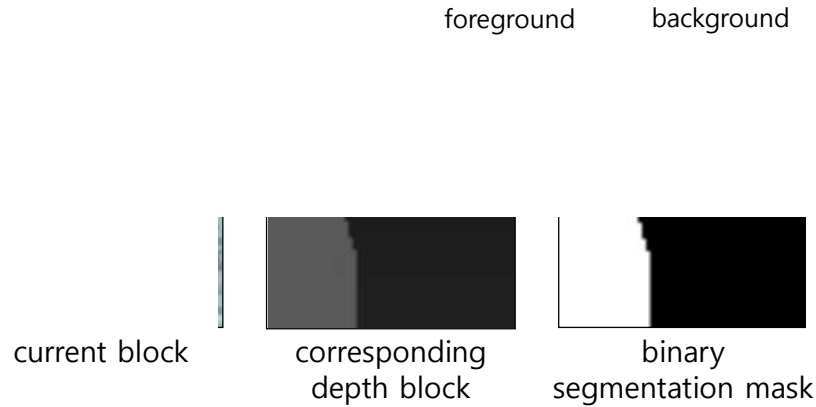


그림 9. 이진 세그멘테이션 마스크 생성 방법
Fig. 9. Generation of binary segmentation mask [25]

the binary segmentation mask.

For signaling the motion information of two segments; the foreground and the background regions, the depth-based block partition mode only allows when the partition mode of the current block is PART_2NxN or PART_Nx2N modes [27].

After the motion or disparity compensation prediction for the foreground and the background regions, the boundaries between two segments are filtered since the region near the boundaries might have the distortion [28].

The DBBP mode needs to fetch two depth blocks for identifying the corresponding depth block used to generate the segmentation mask in addition to the motion or disparity compensation, and thus the DBBP mode is only allowed in which block size is greater than 8x8. As a result, the worst case memory bandwidth is much less than the worst case of HEVC [29].

V. Experimental Results

In order to examine the coding performance of 3D-HEVC,

coding efficiency comparisons with the simulcast approach using HEVC and the multi-view extension of HEVC (MV-HEVC) only allowing the inter-view prediction are carried out based on the common test conditions defined in JCT-3V [30] and the 3D-HEVC reference software 3D-HTM 14.0 [31]. The Bjøntegaard Delta bit-rate (BD-BR), which means an amount of the average bit-rate changes, [32] was used to evaluate the coding performance.

Table 1 shows the coding performance comparison between 3D-HEVC and the HEVC simulcast configuration. “Coded views” indicates the performance of the texture images, and “Synthesized views” indicates the performance of the synthesized images generated by the view synthesis algorithm. As can be seen in Table 1, 3D-HEVC provides the BD-BR performance of 47.2% and 55.1% for the coded and the synthesized views, respectively. Table 2 shows the coding performance between 3D-HEVC and MV-HEVC. As can be seen in Table 2, 3D-HEVC provides the BD-BR performance of 13.4% and 17.6% for the coded and the synthesized views, respectively.

표 1. 3D-HEVC와 HEVC simulcast의 BD-BR 성능 비교
 Table 1. Comparison of BD-BR results between 3D-HEVC and HEVC simulcast

Sequence	Coded views	Synthesized views
Balloons	-43.7%	-45.8%
Kendo	-44.5%	-48.0%
Newspaper1	-37.6%	-51.5%
GT_Fly	-57.3%	-63.0%
Poznan_Hall2	-39.3%	-51.7%
Poznan_Street	-44.5%	-47.9%
Undo_Dancer	-50.8%	-68.1%
Shark	-59.8%	-64.7%
Average	-47.2%	-55.1%

표 2. 3D-HEVC와 MV-HEVC의 BD-BR 성능 비교
 Table 2. Comparison of BD-BR results between 3D-HEVC and MV-HEVC

Sequence	Coded views	Synthesized views
Balloons	-17.8%	-17.5%
Kendo	-18.5%	-19.3%
Newspaper1	-11.9%	-18.4%
GT_Fly	-12.2%	-16.9%
Poznan_Hall2	-13.4%	-17.4%
Poznan_Street	-8.2%	-11.6%
Undo_Dancer	-11.7%	-17.9%
Shark	-13.9%	-21.4%
Average	-13.4%	-17.6%

Table 3 shows the coding performance of each inter-component algorithm described in Section IV. The in

표 3. 3D-HEVC 인터-컴포넌트 알고리즘들의 BD-BR 성능
 Table 3. BD-BR performance of inter-component algorithms in 3D-HEVC

Algorithm	Coded views	Synthesized views
IVMP	-6.6%	-5.7%
VSP	-0.5%	-0.3%
IC	-0.7%	-0.5%
RP	-1.2%	-0.9%
DBBP	-0.5%	-0.3%
Total	-9.5%	-7.7%

ter-component algorithms provide the BD-BR performance of 9.5% and 7.7% for the coded and the synthesized views, respectively. Compared to the comparison results between 3D-HEVC and MV-HEVC, it can be seen that most of the coding gain comes from inter-component coding tools especially for the coded views.

A memory complexity analysis for inter-component algorithms is also conducted based on the method used in [33]. The worst case of HEVC occurs when the bi-predictive 8x8 block is applied. Table 4 shows the worst case of each coding tools in terms of external memory bandwidth requirement. The worst case memory bandwidth of IVMP and IC is 110% to 114% depending on the used memory pattern compared to HEVC, because a depth block fetch is needed in addition to the motion compensation prediction. For VSP, the worst case memory bandwidth is 50% to 81%, since VSP is disallowed in the bi-predictive block. For RP, the worst case memory bandwidth is 87% to 94% since RP uses bi-linear interpolation filter in the motion compensation and the RP for chroma is disabled when the current block is 8x8 block size. For DBBP, the worst case memory bandwidth is 79% to 69% since DBBP is allowed when the block size is greater than 8x8.

표 4. 메모리 패턴에 따른 외부 메모리 대역 요구사항 분석
 Table 4. Analysis of external memory bandwidth requirement in the worst case depending on memory patterns

Coding tool	1x1 pattern	4x2 pattern	8x2 pattern
HEVC MC	100%	100%	100%
IVMP	110%	113%	114%
VSP	50%	66%	81%
IC	110%	113%	114%
RP	87%	83%	94%
DBBP	79%	71%	69%

VI. Conclusion

This paper introduced the disparity derivation method and the inter-component coding tools in 3D-HEVC. 3D-HEVC will be able to efficiently deal with the MVD format to support the next generation 3D video display such as a glasses free 3D television and FTV. The experimental results demonstrated that the 3D-HEVC is more efficient than the HEVC simulcast approach and MV-HEVC.

참 고 문 헌 (References)

- [1] Tech, G. et al., 3D-HEVC Draft Text 7, JCT3V-K1001, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2015)
- [2] Smolic, A. et al., 3D Video and Free Viewpoint Video - Technologies, Applications and MPEG Standards, in Proc. IEEE Int. Conf. on Multimedia and Exposition (2006)
- [3] Smolic, A. et al., Multi-view Video plus Depth (MVD) Format for Advanced 3D Video System, JVT-W100, Joint Video Team (JVT) of ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6 (2007)
- [4] Fehn, C., Depth-Image-Based Rendering (DIBR), Compression and Transmission for a New Approach on 3D-TV, in Proc. the SPIE Stereoscopic Displays and Virtual Reality Systems XI (2004)
- [5] Advanced Video Coding for Generic Audiovisual Services, ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 Part 10 AVC), Ver. 8 (2007)
- [6] Bross, B. et al., High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS & Consent), JCTVC-L1003, Joint Collaborative Team on Video Coding of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2013)
- [7] Zhang, Li. et al., Test Model 11 of 3D-HEVC and MV-HEVC, JCT3V-K1003, Joint Collaborative Team on Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2015)
- [8] Zhang, Li. Et al., CE5.h: Disparity vector generation results, JCT3V-A0097, Joint Collaborative Team on Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2012)
- [9] Sung, J. et al., 3D-CE5.h: Simplification of disparity vector derivation for HEVC-based 3D video coding, JCT3V-A0126, Joint Collaborative Team on Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2012)
- [10] Chen, Y.-W. et al., 3D-CE1.h: Depth-oriented neighboring block disparity vector (DoNBDV) with virtual depth retrieval, JCT3V-C0131,

- Joint Collaborative Team on Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2013)
- [11] Lee, J. Y. et al., 3D-CE2.h related results on disparity vector derivation, JCT3V-C0097, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2013)
 - [12] Park, M. W. et al., 3D-CE2.h related: Simplified NBDV and improved disparity vector derivation, JCT3V-E0142, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2013)
 - [13] Park, M. W. et al., 3D-CE2.h related: Default disparity vector derivation, JCT3V-D0112, Joint Collaborative Team on Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2013)
 - [14] Lee, J. Y. et al., Vertical DV restriction after depth-based refinement, JCT3V-G0074, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2014)
 - [15] Park, M. W. et al., 3D-CE2.h related: Clipping in depth-based disparity vector derivation, JCT3V-E0141, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2013)
 - [16] Park, M. W. et al., 3D-CE2 related: Simplification of DV Derivation for Depth Coding, JCT3V-G0074, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2014)
 - [17] An, J. et al. 3D-CE3: Sub-PU level inter-view motion prediction, JCT3V-F0110, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2013)
 - [18] Park, M. W. et al., 3D-CE1: Results on Adaptive Disabling Inter-view Motion Vector Candidates, JCT3V-H0070, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2014)
 - [19] Tian, D. et al., CE1.h: Backward View Synthesis Prediction using Neighboring Blocks, JCT3V-C0152, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2013)
 - [20] Shimizu, S. et al., 3D-CE1.h: Adaptive block partitioning for VSP, JCT3V-E0207, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2013)
 - [21] Mishurovskiy, M. et al., 3D-CE2.a results on inter-view coding with adaptive luminance compensation, JCT3V-B0031, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2012)
 - [22] Liu, H. et al., 3D-CE2.h: Results of Illumination Compensation for Inter-View Prediction, JCT3V-B0045, , Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2013)
 - [23] Park, M. W. et al. 3D-CE4: Results on IC and ARP Flags Signaling, JCT3V-G0072, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC

- 29/WG 11 (2014)
- [24] Zhang, L. et al. CE4: Advanced residual prediction for multiview coding, JCT3V-D0177, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2013)
- [25] Jäger, F. et al. CE3: Results on Depth-based Block Partitioning (DBBP), JCT3V-G0106, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2014)
- [26] Park, M. W. et al. Simplification of Threshold Derivation for DBBP and DMM4, JCT3V-I0076, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2014)
- [27] Park, M. W. et al. Partition Derivation for DBBP, JCT3V-I0077, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2014)
- [28] Lee, J. Y. et al. Partition boundary filtering in DBBP, JCT3V-H0104, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2014)
- [29] Park, M. W. et al. Memory Complexity for DBBP and VSP, JCT3V-I0078, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2014)
- [30] Rusanovskyy, D. et al, Common Test Conditions of 3DV Core Experiments, JCT3V-G1100, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2014)
- [31] 3D-HEVC reference software version HTM 14.0 https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSsoftware/tags/HTM-14.0/
- [32] Bjontegaard, G., Calculation of Average PSNR Differences between RD-curves, VCEG-M33, ITU-T SG16 Q.6. (2001)
- [33] Alshina, E. et al. BoG Report on SHVC complexity assessment, JCTVC-M0455, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 (2013)

저 자 소 개



박민우

- 2003년 02월 : 경희대학교 컴퓨터공학 학사
- 2005년 02월 : 경희대학교 컴퓨터공학 석사
- 2009년 02월 : 경희대학교 컴퓨터공학 박사
- 2009년 03월 ~ 현재 : 삼성전자 DMC연구소 책임연구원
- ORCID : <http://orcid.org/0000-0001-8489-8863>
- 주관심분야 : 비디오 압축 기술 및 표준화, 영상처리, 멀티미디어 통신



이진영

- 2006년 02월 : 상군관대학교 정보통신공학 학사
- 2008년 08월 : 한국과학기술원 전기전자공학 석사
- 2008년 08월 ~ 현재 : 삼성전자 DMC연구소 책임연구원
- ORCID : <http://orcid.org/0000-0002-4384-2633>
- 주관심분야 : 영상처리, 비디오 압축 기술 및 표준화, 3D 비디오 시스템



김찬열

- 1998년 02월 : 연세대학교 전파공학 학사
- 2003년 02월 : 연세대학교 통신공학 석사
- 2009년 09월 : 더블린대학교 전기전자 박사
- 1998년 02월 ~ 현재 : 삼성전자 DMC연구소 수석연구원
- ORCID : <http://orcid.org/0000-0002-2337-5177>
- 주관심분야 : 비디오 압축 기술 및 표준화, 영상처리