Reusable HEVC Design in 3D-HEVC

Young Su Heo, Gun Bang, and Gwang Hoon Park

This paper proposes a reusable design for the merging process used in three-dimensional High Efficiency Video Coding (3D-HEVC), which can significantly reduce the implementation complexity by eliminating duplicated module redundancies. The majority of inter-prediction coding tools used in 3D-HEVC are utilized through a merge mode, whose extended merging process is based on built-in integration to completely wrap around the HEVC merging process. Consequently, the implementation complexity is unavoidably very high. To facilitate easy market implementation, the design of a legacy codec should be reused in an extended codec if possible. The proposed 3D-HEVC merging process is divided into the base merging process of reusing HEVC modules and reprocessing process of refining the existing processes that have been newly introduced or modified for 3D-HEVC. To create a reusable design, the causal and mutual dependencies between the newly added modules for 3D-HEVC and the reused HEVC modules are eliminated, and the ineffective methods are simplified. In an application of the proposed reusable design, the duplicated reimplementation of HEVC modules, which account for 50.7% of the 3D-HEVC merging process, can be eliminated while maintaining the same coding efficiency. The proposed method has been adopted as a normative coding tool in the 3D-HEVC international standard.

Keywords: HEVC, H.265, 3D-HEVC, merge mode, reusable design, implementation complexity.

I. Introduction

User demand for realistic media is rapidly growing for a wide range of applications, such as ultra high definition video, three-dimensional (3D) video, and a combination of the two. In keeping with this trend, research on new technologies for multi-view video coding is continuously ongoing. The Moving Picture Experts Group (MPEG) and Video Coding Experts Group have been formulating international standardizations of 3D video in the form of extension designs based on the legacy codecs targeted toward 2D video.

One of the most important factors in multi-view video data compression is exploiting inter-view redundancy. Information contained in pictures that have the same time instance but are taken from different views can be critically referenced for dependent-view coding, and thus should be carefully exploited [1]. The first attempt at this was disparity compensated prediction (DCP). DCP performs predictive coding for a current image based on a picture taken from a different view that has already been coded but is located at the same time instance. From the perspective of syntax extension, it is sufficient to change the high-level syntax without changing the low-level syntax, semantics, and decoding process for a block-level coding tool. DCP was inceptively adopted in Multi-view Video Coding (MVC) [2], which is an extension of Advanced Video Coding /H.264 [3], and the HEVC extension to multiple views (MV-HEVC) [4], which is an extension of High Efficiency Video Coding (HEVC)/H.265 [5]. These extension standards have been called *frame-compatible* formats. However, the extension codec designed in a frame-compatible format cannot sufficiently exploit inter-view redundancy. Nevertheless, MVC had been limited to the extension of high-level syntax because of the otherwise heavy increase in implementation complexity of low-level coding tools. A real example is as follows: Throughout the period of MVC

Manuscript received Dec. 11, 2015; revised Apr. 19, 2016; accepted May 16, 2016.

Young Su Heo (medist@khu.ac.kr) and Gwang Hoon Park (corresponding author, ghpark@khu.ac.kr) are with the Department of Computer Engineering, Kyung Hee University, Seoul, Rep. of Korea.

Gun Bang (gbang@etri.re.kr) is with the Broadcasting Media Research Laboratory, ETRI, Daejeon, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.16.2615.0019

standardization, block level coding tools such as illumination compensation (IC) [6], the deblocking filter for IC [7], and motion skip [8] were adopted through core experiments. However, they were ultimately excluded from the finalized MVC profile because of the traditional concerns raised regarding the implementation complexity of macroblock-level coding tools [9].

When a low-level coding tool is newly introduced or modified, two kinds of changes occur in terms of complexity. First, while the coding efficiency can be enhanced, the computational complexity of the algorithm itself increases. However, this cannot be a major reason for excluding all low-level coding tools from the extension design because it is a conventional tradeoff in video coding. One of the major reasons for this is the hardware (H/W) implementation complexity. Because the adoption or modification of a low-level coding tool generally involves changing the internal processes of the coding tool, the existing process of a legacy codec cannot be reused as-is. In other words, not only coding processes that are newly added or modified in an extension codec but also most of the unmodified legacy codec processes have to be newly implemented repeatedly. This greatly increases the cost for H/W implementation in comparison with a frame-compatible format. For this reason, the implementation complexity is also considered as important as the coding efficiency and computational complexity for extension codecs.

However, a recent extension codec targeted toward 3D video has shown a different trend from the existing extension codecs previously discussed. 3D-HEVC [1], which was completed in Feb. 2015, is an example of an extension design that enables realistic 3D navigation in a multi-view environment and is based on HEVC/H.265. It adopts block-level, that is, coding unit (CU), coding tools that exploit not only the inter-view redundancy of multi-view video but also the inter-component dependence between a texture image and its corresponding depth map. However, the efficiency of this coding scheme needed to be improved to overcome the shortcoming in which the legacy codec designs are unable to be reused as-is. It was recently reported that 3D-HEVC can achieve bitrate reductions of 17.6% and 55.1% for the same quality of reconstructed video in comparison with MV-HEVC and HEVC/H.265 simulcast, respectively [10]. However, despite the improved coding efficiency, heavy implementation complexity remains a problem to be addressed. Although, unlike MV-HEVC, reusing designs at the frame-level without any modification may not be possible, extension designs should focus on making each module activated for a low-level coding tool reusable. In the existing 3D-HEVC [11], newly added modules that belong to the extension part of an enhancement layer codec are designed to completely wrap HEVC in a unit of each CU. Consequently, when implementing the HEVC modules for 3D-HEVC as H/W, both the CU-level design and implementation need to be newly redone.

The majority of coding tools newly introduced or modified in 3D-HEVC are utilized through a merge mode [11]; thus, reusable HEVC modules can be found most frequently in the merging process. The merge mode that inherits the motion data of a neighboring block is one of the core technologies that greatly increase the coding performance of HEVC [12] and has become more complicated and sophisticated in 3D-HEVC as the number of views increases. However, even though the existing merging process, which was adopted in the 3D-HEVC [11], had many parts that overlap with HEVC, because it was designed primarily to achieve the maximum coding performance without consideration of the module reusability, the implementation complexity was very high.

This paper proposes a reusable design that can increase the reusability of a legacy codec and reduce the H/W implementation complexity for the merge mode of 3D-HEVC.

II. Proposed Design for Extension of Legacy Codec

Extension codecs are usually designed based on a layered approach using base and enhancement layers. There are two kinds of methodologies when designing a low-level decoding process for extension codecs. Figure 1 shows a data-flow diagram for the decoding process of an extension codec where the enhancement layer codec is designed by extending the base layer codec (that is, a legacy codec). The bitstream transferred from the encoder is separated into the base and enhancement layer components by the de-multiplexer (DEMUX) and then input into the base and enhancement layer decoders, respectively. After the decoding process is conducted based on the syntax restored through the parsing process of the decoder belonging to each layer, each component becomes reconstructed.

As shown in Fig. 1(a), the first extension design concept for the enhancement layer codec is built-in integration. This method either adds new necessary processes to extend the functions of the enhancement layer decoder directly into the legacy decoding process or modifies it. Therefore, it is designed as an alternative decoding process that is repeatedly re-implemented in a manner that completely wraps around the legacy decoding process. Because this method adds new functions into various optimum locations inside the entire decoding process of the legacy codec, it generates the best outcome in terms of coding efficiency. Built-in integration is used for general extension designs and has been applied to most parts of low-level designs for 3D-HEVC [11]. However, the weakness of this method is its high implementation

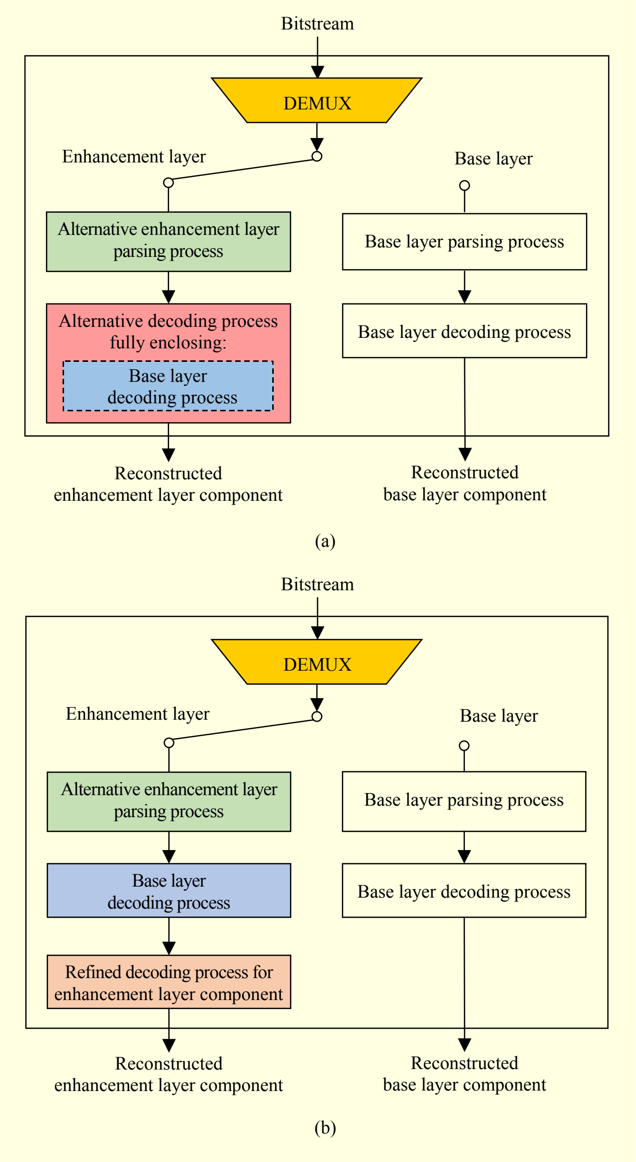

Fig. 1. Data-flow diagram of decoding process in extension codec: (a) built-in integration design and (b) reusable design (proposed codec design).

complexity. As discussed in Section I, because the implementation complexity is a sensitive issue in the extension design, applying built-in integration throughout the entire decoding process cannot be considered a good design.

To solve the implementation complexity problem of built-in integration, this paper proposes a reusable design, which is illustrated in Fig. 1(b). In the decoding process of the enhancement layer of the proposed design, the legacy decoding process of the base layer is first performed as-is. The results, along with the functions newly added in the extended design, are then reprocessed in the refined decoding process for the enhancement layer component. Unlike a built-in integration, the new or modified processes are not added into several locations inside the legacy decoding process. Thus, the low-level design of the legacy decoder becomes reusable. The major task of the succeeding refined decoding process is handling new intrinsic functions for the extension design. This design has the following merits: (1) The H/W implementation complexity is significantly reduced compared with built-in integration because the legacy codec design is not repeated, and (2) the decoding steps prior to the reprocessing are rapidly processed because the branches that assess whether or not the additional functions of extended tools are executed are not placed in various locations within the legacy decoding process. To apply a reusable design to an extension codec, the dependences between the legacy codec and the new functions of the extension codec must be removed. When there is a dependent relationship between HEVC and the extended part of 3D-HEVC, the process for dependency removal should be implemented in the design of the refined decoding process for the enhancement layer.

III. Overview of Existing Merge Mode in 3D-HEVC

The merging process of HEVC is designed to use the motion data inherited from one of the spatiotemporal neighboring prediction units (PUs) without modification. The PU to which the merge mode is applied receives only a candidate index and residual data from the decoder. The following presents an overview of the HEVC merge mode [12] as well as the existing 3D-HEVC merge mode [11], which consists of newly added and modified merging procedures for accepting 3D candidates.

1. Merge Mode in HEVC

A. Merge Candidates

The merge candidates used in the HEVC merging process comprise spatial and temporal candidates. Spatial candidates are located near the current PU that is being coded. As shown in Fig. 2, they can be derived from the information of the previously coded neighboring PUs (that is, the individual motion data located in the lower-left (A_0), left (A_1), upper-right (B_0), upper (B_1), and upper-left (B_2) areas of the figure) [12]. Temporal candidates are derived from the temporally collocated PU that is already coded. They use the motion data derived from both Col_1, which is the same location as the temporally collocated PU of the current PU, and Col_0, which points toward the lower-right of the temporally collocated PU.

B. Merge Candidate List Construction

The merge candidate list (MCL) construction of HEVC and its major block diagrams are overviewed in [12]. The number

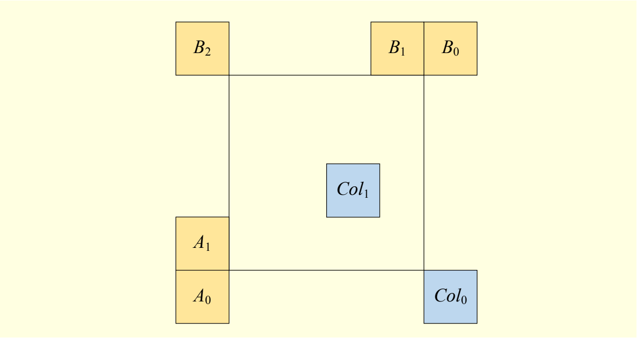

Fig. 2. Positions of spatial and temporal merge candidates for current PU.

of slots of the MCL is normally five. In MCL construction, the first step is to insert the spatial candidates, illustrated in Fig. 2, into the MCL. The insertion priority of the spatial candidates is A_1, B_1, B_0, A_0, and B_2 in that order. Because the motion data of spatial neighboring blocks are very similar, the motion data of the merge candidates derived from them can be overlapped. Therefore, before inserting each candidate into the list, pruning has to be performed when the motion data of a candidate to be inserted are confirmed to be the same as the data of an already inserted candidate through a redundancy check. The second step is to insert a temporal candidate (either Col_1 or Col_0) into the MCL. If the location of each candidate is in an uncoded area or coded in an intra-mode, the motion data cannot be derived from that location; it is possible, therefore, that the MCL cannot be filled. In this case, the empty slots can be filled with virtual candidates [14]. B-slice has two reference picture lists ($L0$ and $L1$), and the included PU performs bi-prediction by using two sets of motion data obtained from $L0$ and $L1$. The first type of virtual candidates is a combined bi-predictive candidate that combines $L0$ and $L1$ motion data of the spatiotemporal candidates inserted into the MCL. If empty slots still remain in the MCL, they are filled with the second type, which is a zero candidate [11], [12].

2. Merge Mode in 3D-HEVC

A. Additional 3D Candidates

In 3D-HEVC, where the corresponding depth map is coded together with the view image as additional information for the synthesis of virtual views, newly added redundancies arise because the characteristics of 3D video coexist along with the spatiotemporal redundancy of 2D video.

a. Texture candidate: For the same view, the motion characteristics between a texture and its corresponding depth map are very similar at the same time instance [11]. The motion data of a texture can be utilized as a predicted value for depth map coding. This methodology, which exploits inter-component dependence, is called motion parameter inheritance (MPI) [11]. Texture candidates need to use MPI in merge mode and are derived through the inheritance of the motion data held by the texture PU.

b. Inter-view disparity candidate: The same arbitrary object in one scene is shown in different locations within dependent-view images and the base-view image at the same time instance because of the different viewpoints. This location difference can be compensated for with a disparity vector (DV). If a DV can be used, it is used as an inter-view disparity candidate (I*v*DC) [13].

c. Inter-view motion candidate: The inter-view motion candidate (I*v*MC) is necessary when using the inter-view motion parameter prediction [14] in 3D-HEVC merge mode. I*v*MC is derived through the inheritance of motion data of the base-view picture at the same instance as the current dependent-view picture. If available, I*v*MC inherits the motion data of its corresponding block and uses them as motion data of the current PU [14].

d. Shifted inter-view motion and shifted inter-view disparity candidates: The shifted inter-view motion candidate (I*v*MCShift) and shifted inter-view disparity candidate (I*v*DCShift) [15] are additional candidates that can improve the coding efficiency by compensating the accuracy of the DV.

e. VSP candidate: The current picture can be virtually synthesized through texture warping using depth information. In 3D-HEVC, view synthesis prediction (VSP) [16] is utilized as a newly added merge candidate.

B. MCL Construction in 3D-HEVC

Candidates that have performed individual derivation processes are inserted into the MCL in the order of a predefined priority. At this time, a redundancy check is carried out on some candidates before they are inserted in order to identify whether or not they have the same motion data as the already inserted candidates [11]. H/W implementations of this construction in the existing 3D-HEVC merge mode are carried out through the built-in integration design shown in Fig. 3, the procedure of which is as follows. Texture candidates are only used in the depth map. The VSP candidates are used only in the texture images of dependent views. If the MCL is filled up while each step of the procedure is performed in order, this procedure is terminated early. If a candidate to be inserted in each step is unavailable, the corresponding step is skipped.

Step 1: The texture candidate (T) is inserted into the MCL.
Step 2: The *IvMC* is inserted into the MCL.
Step 3: A_1 is inserted into the MCL if different from *IvMC*.
Step 4: B_1 is inserted into the MCL if different from *IvMC*.
Step 5: B_0 is inserted into the MCL.

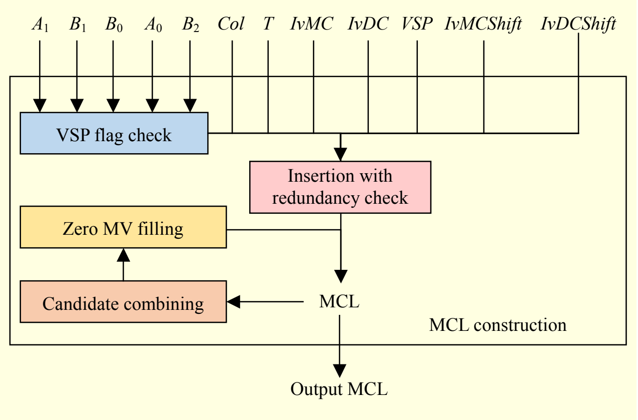

Fig. 3. Built-in integration design of MCL construction in existing merge mode of 3D-HEVC.

Step 6: The *IvDC* is inserted into the MCL if different from both A_1 and B_1.

Step 7: *VSP* is inserted into the MCL.

Step 8: A_0 is inserted into the MCL.

Step 9: B_2 is inserted into the MCL.

Step 10: If *IvMC* is unavailable, or if *IvMC* is available and *IvMCShift* is different from *IvMC*, *IvMCShift* is inserted into the MCL.

Step 11: If *IvMCShift* is unavailable, *IvDCShift* is derived and inserted into the MCL.

Step 12: The temporal candidate (*Col*) is inserted into the MCL.

Step 13: If the current PU belongs to B-slice, combined bi-predictive candidates are derived using the combination pairs and inserted into the empty slots of the MCL.

Step 14: The remaining empty slots of the MCL are filled with zero candidates.

IV. Merge Mode Analysis in Terms of Reusability

As overviewed in Section III, the existing merging process of 3D-HEVC [11] uses built-in integration repeatedly such that all of the decoding processes of the newly added block-level coding tools completely wrap around the functions of the legacy codec (HEVC) without considering the strengths and weaknesses of the extension designs, as explained in Section II.

Figure 4 shows the major process flow of the existing merging process of 3D-HEVC. The green-colored blocks denote the processes that overlap with HEVC [12], whereas the white blocks depict the extended processes for 3D-HEVC. For a reusable design, the processes associated with the additional functions necessary for 3D-HEVC in the built-in integration structure should be separated from the HEVC process and refined. Simplification processes need to be conducted to remove the causal and mutual dependences between the HEVC process and extended part of 3D-HEVC. Thus, a careful dependence analysis should be conducted first. Subsequent to the dependence removal, the extended parts can be separated from the existing built-in integration process and then handled independently.

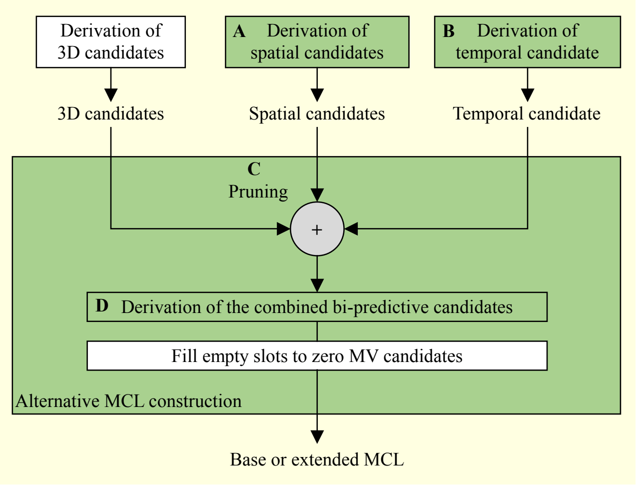

Fig. 4. Existing built-in integration design of merging process in 3D-HEVC.

1. Spatial Candidates

The derivation process of the spatial candidates, marked as **A** in Fig. 4, was partially modified in the existing 3D-HEVC. In HEVC, five spatial candidates are derived in the order of A_1, B_1, B_0, A_0, and B_2. Because the maximum size of the MCL is five, the number of usable spatial candidates is limited to four so as to secure a slot for the temporal candidate after the spatial candidates have been inserted. In the existing 3D-HEVC, however, the maximum size of the MCL had been increased to six. Thus, the number of spatial candidates was not limited to four [11]. The following procedure describes the derivation process of B_2, which is the rear part of the derivation process for spatial candidates. Whereas all of the steps are applicable in HEVC, step 3 was omitted from the existing 3D-HEVC. Because step 3 of the HEVC process is simply omitted in 3D-HEVC, there is no dependence with its extended parts. However, if not limiting the number of spatial candidates has little impact on the coding efficiency, the original HEVC process can be reused as-is.

Step 1: If the motion data of the above-left location of the current PU are valid, they are inherited, and B_2 is set to "available." If not, B_2 is set to "unavailable."

Step 2: If B_2 is available and the motion data of B_2 are the same as either A_1 or B_1, B_2 is set to "unavailable."

Step 3: If B_2 is available and A_1, B_1, B_0, and B_0 are all usable, B_2 is set to "unavailable."

In addition, the derivation process of spatial candidates in the existing 3D-HEVC was extended because the 3D candidate derivation process of the VSP mode was added. Each derived spatial candidate had to be checked to determine whether it was coded in VSP mode. If so, the VSP flag of the corresponding spatial candidate was set to "*true*." When a PU is reconstructed using spatial candidates whose VSP flag is "*true*," the synthesized virtual block instead of the real coded block is used as the predicted block [11]. The extended process of checking the VSP flags in 3D-HEVC can be carried out in the subsequent reprocessing process, which is separated from the spatial candidate derivation process of HEVC.

2. Temporal Candidates

In the derivation process of the temporal candidate, marked as **B** in Fig. 4, motion data are derived from the collocated PU of the reference picture as signaled by its slice header. Because inter-view prediction is also utilized in 3D-HEVC, the types of reference pictures for the current PU and the temporally collocated PU corresponding to the current PU can be either a temporal reference picture or inter-view reference picture. If the reference pictures of the two PUs (that is, the current PU and temporally collocated PU) are different (see the following two cases), the motion data derived from the collocated PU should be additionally processed to be used as a temporal candidate.

Case 1: The reference picture of the current PU is a temporal reference picture, and the reference picture of the temporally collocated PU corresponding to the current PU is an inter-view reference picture.

Case 2: The reference picture of the current PU is an inter-view reference picture, and the reference picture of the inter-view collocated PU corresponding to the current PU is a temporal reference picture.

If the conditions for either case 1 or 2 are met among the pictures included in the reference picture list of the current PU, the index of the first picture identical to the type of reference pictures (temporal or inter-view) of the collocated block is applied to modify the reference picture index of the derived motion data. Because the newly added procedure in the 3D-HEVC temporal candidate derivation process is mutually dependent with the temporal candidate of HEVC, if the dedicated process applied only to 3D-HEVC is separated and reprocessed through refinement, the additional computation and memory access necessary to perform this refining process become a rather severe overhead. Therefore, when altering the 3D-HEVC merging process into a reusable design, using the existing method of 3D-HEVC for the temporal candidate derivation process without any change can rather reduce the implementation cost.

3. Redundancy Checks

There is a causal dependence on the redundancy check process, which is a step prior to the insertion of each candidate into the MCL. As indicated by "Pruning" (marked as **C**) in Fig. 4, redundancy checks between the candidates used in HEVC and the newly introduced 3D candidates have been added in 3D-HEVC. Therefore, to separate the process of inserting 3D candidates into the MCL for refinement, the causal dependence that exists in the redundancy check process of 3D-HEVC should be eliminated.

4. Combined Bi-predictive Candidates

There exists a mutual dependence in the derivation process of combined bi-predictive candidates, marked as **D** in Fig. 4. Candidates that can be combined after insertion into the MCL can be either HEVC or 3D candidates. Mutual dependence exists because these candidates can be combined with each other and one cannot be separated from the other. In addition, the existing 3D-HEVC made the new decision that candidates whose VSP flag is "*true*" cannot be used for combinations [11]. The coded PU with VSP mode is inappropriate for use in combinations because it only has DVs. Hence, the combinations that include VSP candidates or spatial candidates coded with VSP mode are not used to derive combined bi-predictive candidates. Therefore, under the condition that the VSP flags of spatial candidates (which are considered as HEVC candidates) have to be set in advance, there exists a causal dependence. The dependences stated above should be eliminated in order to refine the process of deriving combined bi-predictive candidates as independent and possessing 3D candidates only. In the existing 3D-HEVC, eight combination pairs were newly added to derive the combined bi-predictive candidates. If the eight newly added combination pairs in the 3D-HEVC do not affect the overall coding efficiency, the derivation process of HEVC, which uses only 12 combination pairs, can be reused as-is.

V. Proposed Reusable Merging Process

Based on the design concept introduced in Section II, this section proposes a reusable 3D-HEVC merging process that can reuse the MCL construction of HEVC as-is [17]. The proposed method has been adopted as a normative tool for the 3D-HEVC international standard, and included in the 3D-HEVC Final Draft Amendment (FDAM 4) [18] and its recent HTM-14.0 reference software (S/W) [19]. The block diagram of the proposed method is illustrated in Fig. 5.

The white blocks indicate that the original processes of HEVC are reused, whereas the green-colored blocks denote

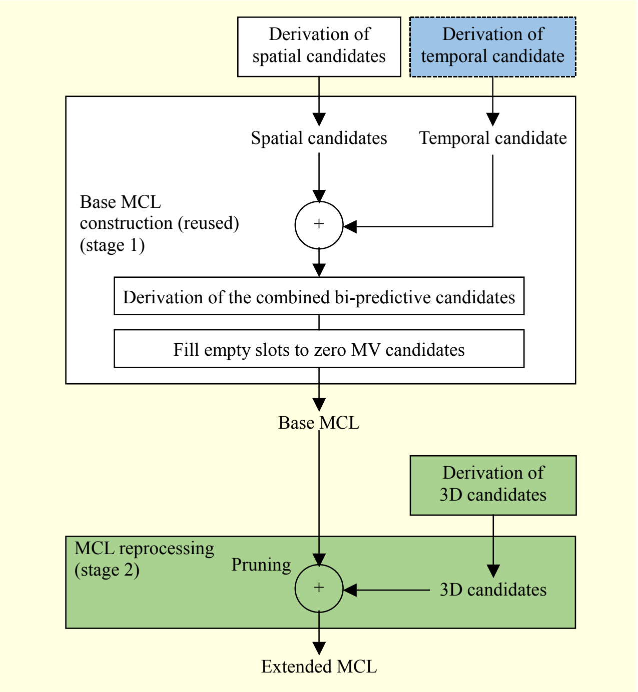

Fig. 5. Proposed reusable design for merging process in 3D-HEVC.

refined processes for the newly added 3D candidates in 3D-HEVC. Among the existing built-in integration processes in Fig. 4, original HEVC processes such as the derivation process of spatial candidates, MCL construction process, and derivation process of combined bi-predictive candidates are reused as they are in the base MCL construction process (stage 1). To apply the proposed reusable design, the simplifications and refinements suggested to solve the issues of causal and mutual dependences of the existing merging process of 3D-HEVC are carried out in the MCL reprocessing process (stage 2). For the derivation process of temporal candidates, the existing built-in design of 3D-HEVC is maintained as indicated by the dotted lines and blue color.

1. Simplification and Dependence Removal for Derivation of HEVC Merge Candidates

A. Derivation Process of Spatial Candidates

The derivation process of the spatial candidates used in HEVC is reused without any modification. The priority of the B_2 candidate is ninth in the MCL construction procedure of the existing merge mode in 3D-HEVC (step 9 in the procedure of Section III). Table 1 lists the experimental results for the measurement of the selection probability of each merge candidate. Because candidates with a high priority have the optimal RD performance, the B_2 candidate is very unlikely to be selected (0.17% in a dependent texture) as a merge index even if it is inserted into the MCL. Because the B_2 candidate has little impact on the coding efficiency, the HEVC process, which limits the availability of the B_2 candidate, can be maintained without any changes.

Table 1. Selection probability (%) for merge candidates.

Candidates	Selection probability		Candidates	Selection probability	
	Dependent texture	Depth		Dependent texture	Depth
T	N/A	85.09	A_0	0.23	0.11
IvMC	57.04	N/A	B_2	0.17	0.56
A_1	19.23	6.61	*IvMCShift*	1.75	N/A
B_1	10.72	3.78	*IvDCShift*	0.49	N/A
B_0	2.23	0.99	*Col*	0.90	1.61
IvDC	4.39	N/A	Bi	0.31	0.77
VSP	2.48	N/A	Zero	0.31	0.48

B. Derivation Process of Combined Bi-predictive Candidates

The following three simplification processes are applied to the proposed reusable design to derive combined bi-predictive candidates.

a. The modifications to the derivation process of the combined bi-predictive candidates listed in Section IV have no negative impact on the coding efficiency of 3D-HEVC, as indicated in Table 6. The eight combination pairs newly added in 3D-HEVC are actually only used when five candidates are inserted into the MCL prior to the execution of the combining process, which is unlikely to occur, as indicated in Table 5. Therefore, the proposed method simplifies this process by using intact combination pairs of HEVC as the combination pairs for 3D-HEVC.

b. The process for determining combination pairs based on VSP flags also has little impact on the coding efficiency [17], [20]. In 3D-HEVC, because there are 12 types of candidates that should be inserted prior to the combined bi-predictive candidates, it is extremely rare that many MCL slots still remain empty after these candidates are all inserted. Therefore, the proposed method removes the causal dependence by eliminating the decision process based on the VSP flag check.

c. To eliminate the mutual dependence, which exists because the HEVC and 3D candidates are combined with each other, only HEVC candidates such as spatial and temporal candidates are combined in the proposed method. In most cases, primitive candidates rather than virtual candidates that combine 3D candidates are selected as a merge index. Hence, eliminating the 3D candidates from combination has no negative impact on the overall coding efficiency, as indicated in Table 6.

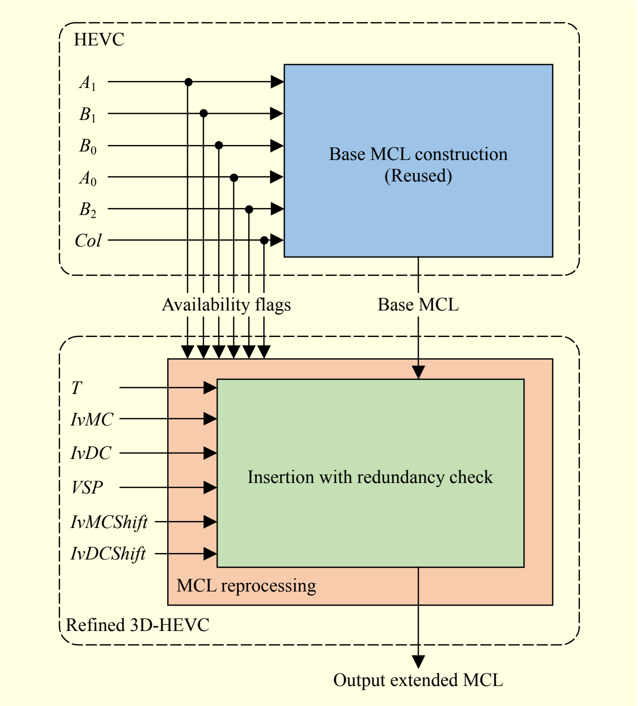

Fig. 6. Block diagram of proposed reusable design for implementation.

2. MCL Reprocessing

The proposed MCL reprocessing process is conceptually illustrated in the lower part of Fig. 5. In stage 1, HEVC modules are reused and processed without any changes, and the base MCL (MCL_B) is produced. In stage 2 (MCL reprocessing), rearranged 3D candidates in addition to the candidates of MCL_B (spatial, temporal, and virtual) are inserted to produce the extended MCL (MCL_E). The reusable design illustrated in Fig. 6 is completely different from the existing built-in integration design of 3D-HEVC illustrated in Fig. 3, and indicates that the MCL construction process of HEVC is reused as-is. The MCL reprocessing illustrated in Figs. 5 and 6 complies with the following procedure. Redundancy checks between HEVC and 3D candidates are carried out in the same manner as in the existing 3D-HEVC. If MCL_E is filled up while the steps of the procedure are performed in order, this procedure is terminated early. If the candidates to be inserted in each step are unavailable, the relevant step is skipped.

Step 1: *T* is inserted into MCL_E.

Step 2: *IvMC* is inserted into MCL_E.

Step 3: A_1 of MCL_B is inserted into MCL_E if different from *IvMC*.

Step 4: B_1 of MCL_B is inserted into MCL_E if different from *IvMC*.

Step 5: B_0 of MCL_B is inserted into MCL_E.

Step 6: *IvDC* is inserted into MCL_E if different from both A_1 and B_1.

Step 7: *VSP* is inserted into MCL_E.

Step 8: A_0 of MCL_B is inserted into MCL_E.

Step 9: B_2 of MCL_B is inserted MCL_E.

Step 10: If *IvMC* is unavailable, or if *IvMC* is available and *IvMCShift* is different from *IvMC*, *IvMCShift* is inserted into MCL_E.

Step 11: If *IvMCShift* is unavailable, *IvDCShift* is derived and then inserted into MCL_E.

Step 12: Among the candidates inside MCL_B, those other than spatial candidates (that is, candidates that do not belong to $\{A_1, B_1, B_0, A_0, B_2\}$) are inserted into MCL_E in order.

VI. Experimental Results

The experimental conditions strictly followed the common test conditions determined by JCT-3V for 3D-HEVC international standardization [21]. Simulations were conducted for cases involving three views and their corresponding depth images. The coding efficiencies were compared through measurements of the Bjøntegaard-Delta (BD) rate [22], which is based on PSNRs between the reconstructed images from the coded bitstream and the original images. The computational complexities were compared through the runtime for encoding and decoding in a sequential coding environment.

As explained in Section V, the proposed reusable merging process has been implemented in the recent HTM-14.0 reference S/W [14]. In order to obtain finalized results in comparison with the proposed method implemented in HTM-14.0, the existing method, which was overviewed in Section III and had been implemented until HTM-8.2 [23], was re-implemented in HTM-14.0 and then compared.

As criteria for measuring the portion of the HEVC merging process that is integrated in the existing merge mode of 3D-HEVC, the numbers of three major operations, that is, the availability determinations for each candidate, motion data fetches, and redundancy checks, can be used. The availability determination is carried out in each derivation process for a merge candidate. To determine whether it can be used. A motion data fetch is carried out on the merge candidates that directly inherit motion data from other blocks, such as spatial, temporal, *T*, *IvMC*, and *IvMCShift*. It accompanies an external memory access. A redundancy check is another major operation that is performed in the candidate derivation process and MCL construction. Thus, a reduction in the implementation complexity can be identified by measuring the number of executions of these three major operations by comparing the numbers before and after the use of the proposed method for the processing of the 3D-HEVC merge mode.

Table 2. Reduction of major operations on proposed merging process.

Number of operations	Before refinement	After refinement	Refinement rate	Reduction rate
Availability determinations	12	7	58.3%	41.7%
Motion data fetches for dependent texture	8	3	37.5%	62.5%
Redundancy checks	10	5	50.0%	50.0%
Calculations of combined bi-predictive candidate	20	0	eliminated	100.0%

Table 3. Reduction rate for number of motion data fetches of a dependent texture.

Sequences	qp25	qp30	qp35	qp40	Average
Balloons	64.5%	64.4%	64.1%	63.8%	64.2%
Kendo	64.8%	64.6%	64.2%	63.9%	64.4%
Newspaper	64.4%	64.3%	63.9%	63.7%	64.1%
Gt_Fly	64.6%	64.3%	63.8%	63.3%	64.0%
Poznan_Hall2	64.2%	63.7%	63.4%	63.1%	63.6%
Poznan_Street	64.9%	64.5%	63.9%	63.4%	64.1%
Undo_Dancer	65.2%	64.6%	64.1%	63.7%	64.4%
Shark	64.8%	64.4%	64.0%	63.5%	64.2%
Average	64.7%	64.3%	63.9%	63.5%	64.1%

Table 4. Reduction rate for number of redundancy checks.

Sequences	qp25	qp30	qp35	qp40	Average
Balloons	47.3%	46.8%	45.9%	44.8%	46.2%
Kendo	47.8%	47.1%	46.1%	45.0%	46.5%
Newspaper	46.7%	46.2%	45.3%	44.4%	45.6%
Gt_Fly	48.0%	47.2%	45.8%	44.4%	46.4%
Poznan_Hall2	46.4%	45.2%	44.4%	43.8%	45.0%
Poznan_Street	47.9%	46.9%	45.6%	44.4%	46.2%
Undo_Dancer	48.8%	47.7%	46.4%	45.4%	47.1%
Shark	48.3%	47.4%	46.1%	44.9%	46.7%
Average	47.7%	46.8%	45.7%	44.6%	46.2%

Table 5. Percentage of use of existing combination pairs in derivation of combined bi-predictive candidates.

Sequences	qp25	qp30	qp35	qp40	Average
Balloons	96.1%	97.2%	98.3%	99.3%	97.7%
Kendo	93.0%	95.4%	97.1%	98.2%	95.9%
Newspaper	98.6%	99.0%	99.4%	99.6%	99.1%
Gt_Fly	92.4%	92.9%	95.9%	98.4%	94.9%
Poznan_Hall2	98.1%	99.2%	99.7%	99.9%	99.2%
Poznan_Street	97.8%	99.0%	99.6%	99.8%	99.0%
Undo_Dancer	95.5%	96.2%	97.9%	99.0%	97.1%
Shark	90.8%	92.1%	94.7%	97.3%	93.7%
Average	95.3%	96.4%	97.8%	98.9%	97.1%

Table 2 lists the theoretical results before and after the use of the proposed method in the 3D-HEVC merging process for comparison. For the availability determination, because of the derivation processes of reused spatial candidates, the 12 candidates ware reduced to seven and the reduction rate was expected to be 41.7%. Motion data fetches for a dependent texture were performed once for each of the five spatial candidates, one temporal candidate, and *IvMC* and *IvMCShift* of the 3D candidates for a total of eight applications. Because the derivations of the spatial candidates were reused, the number of motion data fetches decreased from eight to three with the proposed method. Therefore, the reduction rate was expected to be 62.5%. As mentioned in Section III-2-B, the number of redundancy checks was ten in total. However, with the proposed method, there is no need to newly implement a redundancy check between spatial candidates in 3D-HEVC. Thus, the number of redundancy checks for the procedures newly implemented in the proposed method was reduced to five because the derivation processes of spatial candidates were reused. The reduction rate was expected to be 50.0%. As explained in Section V, further combination processing is very unlikely to occur after the combination process of combined bi-predictive candidates, which is performed in the MCL construction process of HEVC. Hence, because the combination process of combined bi-predictive candidates, which is included in the MCL construction process of the existing 3D-HEVC, was eliminated, the number of combinations in the extended parts decreased from a total of 20 to zero for a reduction rate of 100%.

Several experiments were conducted to verify whether the theoretically expected reductions coincided with the actual results. Tables 3 through 5 list the results. For the availability determination, because the merging process had to be performed on each candidate, the theoretical, shown in Table 2, and actual numbers of executions were the same. Thus, the proposed reusable design reduced the number of availability determinations by 41.7%. Table 3 lists the experimentally measured reduction in the number of motion data fetches with the reusable design for the merging process of dependent texture images. The experimentally measured average reduction rate (64.1%) was confirmed to be close to the theoretically expected reduction rate given in Table 2.

Table 6. BD-rate comparisons between existing and proposed methods.

Sequences	Inter-view video			Video BD-rate	Synth. BD-rate
	View 0	View 1	View 2		
Balloons	0.00%	0.06%	0.13%	–0.01%	–0.02%
Kendo	0.00%	–0.06%	–0.19%	–0.09%	–0.13%
Newspaper_CC	0.00%	0.06%	–0.02%	0.00%	–0.06%
GT_Fly	0.00%	0.07%	–0.12%	–0.02%	–0.01%
Poznan_Hall2	0.00%	0.15%	–0.24%	–0.03%	–0.06%
Poznan_Street	0.00%	0.03%	0.03%	0.01%	–0.03%
Undo_Dancer	0.00%	0.01%	0.00%	–0.01%	–0.06%
Shark	0.00%	0.13%	0.06%	–0.01%	–0.01%
Average	0.00%	0.06%	–0.05%	–0.02%	–0.05%

Table 7. Encoding and decoding average time results.

Sequences	Encoding time	Decoding time
Balloons	99.9%	100.0%
Kendo	100.0%	100.0%
Newspaper_CC	100.0%	100.0%
GT_Fly	100.0%	100.0%
Poznan_Hall2	99.9%	99.9%
Poznan_Street	100.0%	100.0%
Undo_Dancer	100.0%	100.1%
Shark	100.0%	100.0%
Average	100.0%	100.0%

Table 4 lists the experimental results for the reduction in the number of redundancy checks with the proposed method. In the experiment, the number of redundancy checks was reduced by an average of 46.2%. This reduction rate is close to the theoretically expected reduction rate in Table 2. To confirm the validity of the combination-pair simplification of the combined bi-predictive candidates described in Section V, the percentage of HEVC combination pairs against the total combination pairs was measured during the MCL construction process for a dependent texture. The results of this experiment are listed in Table 5. Most of the combination pairs newly added in 3D-HEVC were not used because 97.1% of the combinations of combined bi-predictive candidates were made only of HEVC combination pairs on average. Therefore, completely eliminating the newly added combination pairs from the proposed method was found to be a valid approach.

Table 6 lists the experimental results confirming the impact of the proposed method on the coding efficiency. When the total (averaged results of views 1 through 3 and their corresponding depth data) and synthesized bitrates were compared, the average BD rates were –0.02% and –0.05% against the existing method, respectively. This confirms that the coding efficiencies of the existing and proposed methods were exactly the same. Table 7 compares the measured computational complexity necessary to apply the proposed and existing methods. The experimental results confirmed that the encoding and decoding computational complexities of the proposed and existing methods were the same.

The above experimental results confirmed that the proposed method can significantly reduce the implementation complexity by cutting back repeatedly used modules in the built-in integration by approximately 50.7% (41.7% in terms of availability determination, 64.1% in terms of motion data fetches of dependent textures, and 46.2% in terms of redundancy checks) while maintaining the same coding efficiency and computational complexity as the existing method.

VII. Conclusion

This paper proposes a reusable design that can reduce the implementation complexity of the 3D-HEVC merge mode. The proposed method is divided into two stages: (1) the HEVC merging process, which is reused as-is, and (2) MCL reprocessing, where only the newly added and modified processes for 3D-HEVC are simplified and refined. The experimental results confirmed that the proposed method is able to reduce the implementation complexity by reducing the duplicated modules by approximately 50.7% while maintaining the same coding efficiency that can be maximally achieved by the existing method.

References

[1] G.J. Sullivan et al., “Standardized Extensions of High Efficiency Video Coding (HEVC),” *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, Dec. 2013, pp. 1001–1016.

[2] A. Vetro, T. Wiegand, and G.J. Sullivan, “Overview of the Stereo and Multiview Video Coding Extensions of the H.264/MPEG-4 AVC Standard,” *Proc. IEEE*, vol. 99, no. 4, 2011, pp. 626–642.

[3] T. Wiegand et al., “Overview of the H.264/AVC Video Coding Standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, July 2003, pp. 560–576.

[4] M. Domański et al., “Coding of Multiple Video + Depth Using HEVC Technology and Reduced Representations of Side Views and Depth Maps,” *Picture Coding Symp.*, Krakow, Poland, May 7–9, 2012, pp. 5–8.

[5] ISO/IEC/JTC 1, *Information Technology—High Efficiency Coding and Media Delivery in Heterogeneous Environments—Part 2: High Efficiency Video Coding*, ISO/IEC 23008-2:2013 (ver. 1), Nov. 2013.

[6] J.H. Hur, S. Cho, and Y.L. Lee, "Adaptive Local Illumination Change Compensation Method for H.264/AVC-Based Multiview Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, Nov. 2007, pp. 1496–1505.

[7] G.H. Park et al., "Deblocking Filtering for Illumination Compensation in Multiview Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 10, 2008, pp. 1457–1461.

[8] H.S. Koo, Y.J. Jeon, and B.M. Jeon, "Motion Information Inferring Scheme for Multi-view Video Coding," *IEICE Trans. Commun.*, vol. E91.B, no. 4, Apr. 2008, pp. 1247–1250.

[9] G.J. Sullivan et al., "Meeting Report of the 26th JVT Meeting," JVT-Z200, Antalya, Turkey, Jan. 2008.

[10] M.W. Park, J.Y. Lee, and C. Kim, "Overview of Inter-component Coding in 3D-HEVC," *J. Broadcast Eng.*, vol. 20, no. 4, July 2015, pp. 545–556.

[11] G. Tech et al., "3D-HEVC Draft Text 1," *JCT3V-E1001*, Vienna, Austria, Aug. 2013.

[12] P. Helle et al., "Block Merging for Quadtree-Based HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, Dec. 2012, pp. 1720–1731.

[13] L. Zhang, Y. Chen, and L. Liu, "3D-CE5.h: Merge Candidates Derivation from Disparity Vector," *JCT3V-B0048*, Shanghai, China, Oct. 2012.

[14] J. An et al., "3D-CE5.h Related: Inter-view Motion Prediction for HEVC-Based 3D Video Coding," *JCT3V-A0049*, Stockholm, Sweden, July 2012.

[15] L. Zhang et al., "Inter-view Motion Prediction in 3D-HEVC," *IEEE Int. Symp. Circuits Syst.*, Melbourne, Australia, June 1–5, 2014, pp. 17–20.

[16] F. Zou et al., "View Synthesis Prediction in the 3-D Video Coding Extensions of AVC and HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 10, Oct. 2014, pp. 1696–1708.

[17] G. Bang et al., "3D-CE3.h: Results on Simple Merge Candidate List Construction for 3DV," *JCT3V-F0093*, Geneva, Switzerland, Oct. 2013.

[18] G. Tech et al., "3D-HEVC Draft Text 7," *JCT3V-K1001*, Geneva, Switzerland, Feb. 2015.

[19] JCT-3V, *Subversion Repository for the 3D-HEVC Test Model Version HTM-14.0*, Aachen, Germany, 2015.

[20] L. Zhang and Y. Chen, "CE3 Related: Combined Bi-predictive Merging Candidates for 3D-HEVC," *JCT3V-F0129,* Geneva, Switzerland, Oct. 2013.

[21] D. Rusanovskyy, K. Müller, and A. Vetro, "Common Test Conditions of 3DV Core Experiments," *JCT3V-G1100*, CA, USA, Jan. 2014.

[22] G. Bjontegaard, "Calculation of Average PSNR Differences between RD-Curves," *Proc. VCEG Meeting*, Austin, TX, USA, Jan. 2001, pp. 1–5.

[23] JCT-3V, *Subversion Repository for the 3D-HEVC Test Model Version HTM-8.2*, Aachen, Germany, 2014.

Young Su Heo received his BS and MS degrees in computer engineering from Kyung Hee University, Seoul, Rep. of Korea in 2010 and 2012, respectively. Since 2012, he has been a PhD candidate in the same department. His research interests include video signal processing, multimedia systems, and computer vision.

Gun Bang received his BS and MS degrees in computer science and engineering from Hallym University, Gwangwon, Rep. of Korea in 1995 and 1997, respectively, and his PhD in computer science and engineering from Korea University, Seoul, Rep. of Korea in 2014. Since 2000, he has been a principal researcher in the Broadcasting and Telecommunications Media Research Lab at ETRI, Daejeon, Rep. of Korea. His research interests include video signal processing, image processing, and computer vision.

Gwang Hoon Park received his BS and MS degrees in electronic engineering from Yonsei University, Seoul, Rep. of Korea in 1985 and 1987, respectively, and his MS and PhD degrees in electrical engineering and applied physics from Case Western Reserve University, OH, USA in 1991 and 1995, respectively. He was a principal research engineer with the Information and Telecommunication Research and Development Center, Hyundai Electronics Industries, Icheon, Rep. of Korea from 1995 to 1997, and an associate professor with the Department of Computer Science at Yonsei University, Wonju, Rep. of Korea from 1997 to 2001. Since 2001, he has been a professor in the Department of Computer Engineering, Kyung Hee University, Yougin, Rep. of Korea. He has co-authored 71 journal papers, is the co-inventor of 304 registered patents, and has also co-authored 115 technical contributions for international standardizations. His research interests include video signal processing, multimedia systems, and computational intelligence.

http://dx.doi.org/10.4218/etrij.16.2615.0019