Overview of Multi-view Video Coding

Yo-Sung Ho and Kwan-Jung Oh
Gwangju Institute of Science and Technology (GIST)
1 Oryong-dong, Buk-gu, Gwangju, 500-712, Korea
Phone: (+82) 62-970 2258 Fax: (+82) 62-970 3164
E-mail: {hoyo, kjoh81}@gist.ac.kr

ABSTRACT

With the advancement of computer graphics and computer vision technologies, the realistic visual system can come true in the near future. The multi-view video system can provide an augmented realism through selective viewing experience. The multi-view video is a collection of multiple videos capturing the same 3D scene at different viewpoints. Since the data size of the multi-view video increases proportionally to the number of cameras, it is necessary to compress multi-view video data for efficient storage and transmission. This paper provides an overview of multi-view video coding (MVC) and describes its applications, requirements, and the reference software model for MVC.

Keywords: multi-view video coding,

1. Introduction

In recent years, various multimedia services have become available and the demands for realistic multimedia systems are growing rapidly. A number of three-dimensional (3D) video technologies, such as holography, two-view stereoscopic system with special glasses, 3D wide screen cinema, and multi-view video have been studied to satisfy these demands.

Among them, the multi-view video is the key technology for various applications, including free-viewpoint video (FVV), free-viewpoint television (FVT), immersive teleconference, and 3DTV. The traditional video is a two-dimensional (2D) medium and only provides a passive way for viewers to observe the scene. However, the multi-view video can offer arbitrary viewpoints of dynamic scenes and thus allows more realistic video to users.

The multi-view video includes multiple video sequences captured by several cameras at the same time, but different locations. Because of the increased number of cameras, the multi-view video contains a large amount of data. Since this system has serious limitations on data distribution applications, such as broadcasting, multimedia streaming services, and other commercial applications, we need to compress the multi-view sequence efficiently without sacrificing its visual quality significantly [1-4].

In the past, MVC has been studied in several video coding standards. The MPEG-2 multi-view profile (MVP) proposed block-based stereoscopic coding to encode the stereo video. Motion-compensated prediction (MCP) has been used to reduce temporal redundancy, and disparity compensated prediction (DCP) has been employed to reduce inter-view redundancy [5]. The MPEG-4 MAC (multiple auxiliary component) is also related to MVC. In addition, H.263 and H.264 are tried for MVC. However, none of them supports MVC efficiently [6, 7].

Recently, ISO/IEC JTC1/SC29/WG11 Moving Picture Experts Group (MPEG) has recognized the importance of MVC technologies, and an ad hoc group (AHG) on 3-D audio and visual (3DAV) has been established since December 2001. Four main exploration experiments (EE) on 3DAV were performed from 2002 to 2004: EE1 on omni-directional video, EE2 on FTV, EE3 on coding of stereoscopic video using multiple auxiliary components (MAC), and EE4 on depth/disparity coding for 3DTV and intermediate view interpolation.

In response to the Call for Comments (CfC) issued in October 2003, a number of companies have expressed their interests for a standard that enables FTV and 3DTV. After MPEG called interested parties to bring evidences on MVC technologies in October 2004 [7], some evidences were recognized in January 2005 and a Call for Proposals (CfP) on MVC has been issued in July 2005. Then, the responses to the Call were evaluated in January 2006.

MPEG MVC group has been moved to the Joint Video Team (JVT) from the April 2006. JVT consists of experts from MPEG and ITU-T SG16 Video Coding Experts Group (VCEG). After the MVC group has been transferred from MPEG to JVT, the standardization is being carried out actively through various core experiments (CE) in 3DAV AHG. A joint multi-view video model (JMVM) has been developed as an extension of the H.264/MPEG-4 AVC to support the development of the future MVC reference software.

This paper provides an overview of MVC. It also describes its applications, requirements, reference software model, and core technologies.

2. Applications of Multi-view Video Coding

2.1 Free Viewpoint Television (FTV)

In multi-view video, the viewpoint and view direction can be interactively changed. During such viewing, the viewers can experience the free viewpoint navigation within the range covered by the shooting cameras. Such a scenario can appear in the following applications [8]:

(a) Entertainment – concert, sports, multi-user game, movie, drama, news
(b) Education – cultural archives, manual with real video, instruction of sports playing, medical surgery
(c) Sightseeing – zoo, aquarium, botanical garden, museum
(d) Surveillance – traffic intersection, underground parking, bank
(e) Archive – space archive, living national treasures, traditional entertainment
(f) Art/Content – creation of new type of media art and digital content

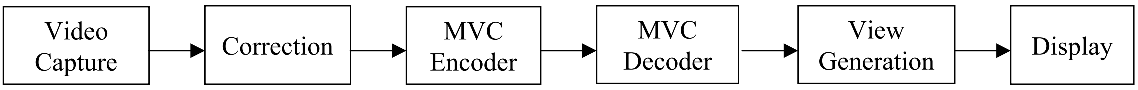

Figure 1: Basic components of the FTV system

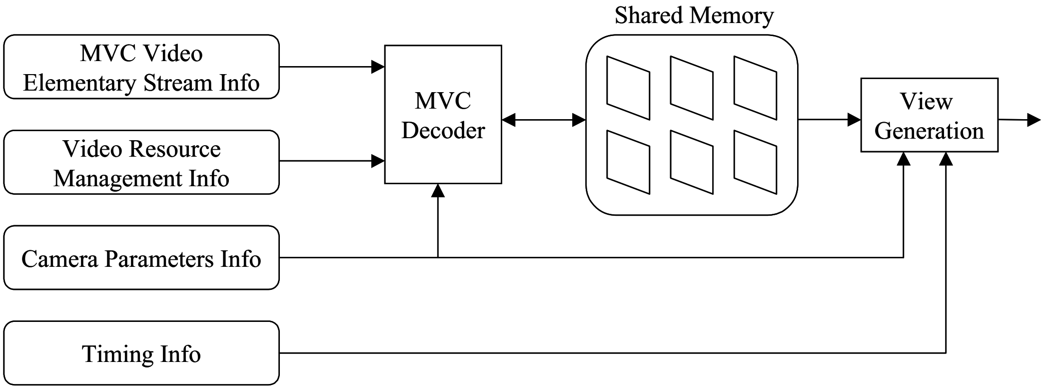

Figure 2: Architecture of the FTV decoder

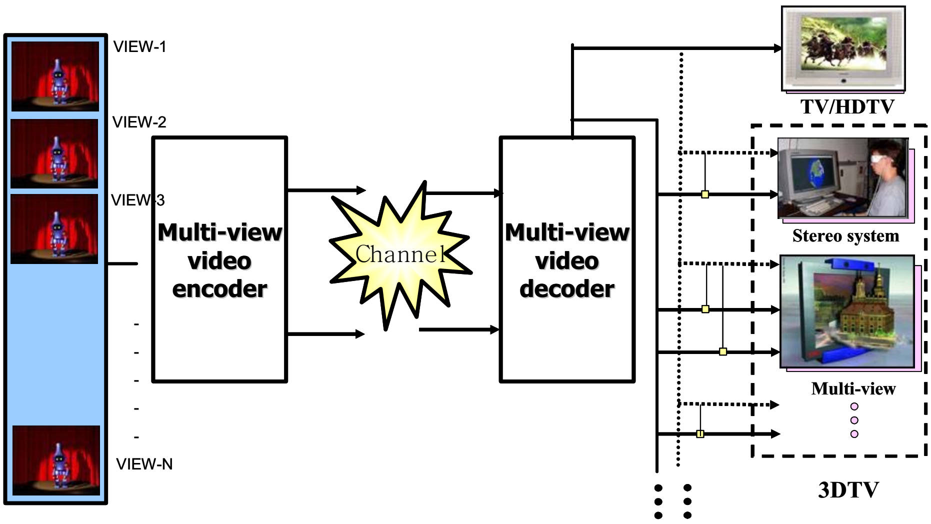

Figure 3: Example of the 3DTV system

Figure 1 depicts the basic components of an example FTV system [8]. The output images from the MVC decoder are used for FTV view generation; this view generation procedure may interpolate images from different views. In order to achieve high-quality view generation results, we may need a correction process, i.e., rectification of misalignment and normalization of colors, in most cases. In the example FTV system shown in Fig. 1, the correction is applied prior to encoding.

Figure 2 shows a more detailed architecture of the FTV decoder. Input streams to the FTV decoder include multi-view video elementary information, video resource management information, timing information, and camera parameters information. In this architecture, the MVC decoder provides reconstructed video data, which is then used in the view generation process. Note that camera parameters may also be used during the MVC decoding process. Video resource management information may be used for managing the picture memory in an efficient way and for generating predictive images for the MVC decoder. Finally, view generation is performed according to the video data information and associated camera parameters information [8].

2.2 Three-dimensional TV (3DTV)

3DTV can be regarded as an extension of the current stereoscopic movie. While all viewers share the same viewpoint in the stereoscopic movie, multiple cameras are used to capture the light field of the scene in 3DTV. When such a light field is displayed, multiple viewers can see different stereoscopic views consistent with their relative locations. However, user interaction from a viewer to the system may not be required. Broadcasting is one of the important applications of 3DTV. By definition, the stereoscopic TV is included in the domain of 3DTV and is expected to be the first mass market application of 3DTV.

Figure 3 shows an example of the 3DTV system. Input videos are multi-view videos that are captured by multiple cameras. The multi-view video encoder encodes and transmits these videos through the channel. End-users may have different types of displays, such as 2D SDTV/HDTV, stereo TV, 3DTV that supports multi-views.

Multi-view videos are captured by various types of multiple cameras, such as 1D parallel, 2D parallel, 1D arc, and so on. They are encoded by the MVC encoder and transmitted through the broadcasting channel. The MVC decoder reconstructs the coded multi-view videos. Because TV viewers can have various types of display systems, the 3DTV system should support the intermediate view reconstruction (IVR) technique for multi-view displays when the number of views to be displayed is larger than the number of encoded views. The standard should also describe the basic relations between the auxiliary data, such as camera parameters, depth, etc., and the basic construction of the interpolated views.

2.3 Immersive Teleconference

In immersive teleconference, there is an interaction between viewers. Participants at different geographical sites meet virtually and see one another in either free viewpoint or 3DTV style. The immersiveness provides a more natural way of communications.

3. Requirements for Multi-view Video Coding

MVC algorithms should satisfy some requirements. In the following, we use "shall" if a certain requirement is mandatory, and "should" if a certain requirement is desirable, but not necessarily required. Requirements for MVC are largely divided into compression related requirements and system support related requirements [8].

3.1 Compression Related Requirements

3.1.1 Compression efficiency

MVC shall provide high compression efficiency relative to independent coding of each view of the same content. Some overhead, such as camera parameters, may be necessary for facilitating view interpolation, i.e., trading coding efficiency for functionality. However, the overhead data should be limited in order to increase acceptance of new services.

3.1.2 View scalability

MVC shall support a scalable bitstream structure to allow for access of selected views with minimum decoding effort. This enables the video to be displayed on a multitude of different terminals and over networks with varying conditions.

3.1.3 Free viewpoint scalability

MVC shall support a scalable bitstream structure to allow for access to partial data from which new views can be generated, i.e., not the original camera views, but the generated views from them. Such content can be delivered to various types of displays. This enables the functionality of free viewpoint navigation on a scalability basis.

3.1.4 Spatial/Temporal/SNR scalability

SNR scalability, spatial scalability, and temporal scalability should be supported.

3.1.5 Backward compatibility

At any instant in time, the bitstream corresponding to one view shall be conforming to AVC.

3.1.6 Resource consumption

MVC should be efficient in terms of resource consumption, such as memory size, memory bandwidth, and processing power.

3.1.7 Low delay

MVC shall support low encoding and decoding delay modes. Low delay is very important for the real-time applications such as a streaming and broadcasting using multi-view video.

3.1.8 Robustness

Robustness to errors, also known as error resilience, should be supported. This enables the delivery of multi-view video contents on error-prone networks, such as wireless networks and other networks.

3.1.9 Resolution, bit depth, chroma sampling format

MVC shall support spatial resolutions from QCIF to HD. MVC shall support the YUV 4:2:0 format. MVC shall support 8 bits per pixel component. Future applications may require higher bit depths and higher chroma sampling formats.

3.1.10 Picture quality among views

MVC should enable flexible quality allocation over different views. For instance, consistent quality might be required for some applications.

3.1.11 Temporal random access

MVC shall support random access in the time dimension. For example, it shall be possible to access a frame at a given time with minimal decoding of frames in the time dimension.

3.1.12 View random access

MVC shall support random access in the view dimension. For example, it shall be possible to access a frame in a given view with minimal decoding of frames in the view dimension.

3.1.13 Spatial random access

MVC should support random access to a spatial area in a picture. This may be treated as a view random access if a view is composed of several spatially smaller views.

3.1.14 Resource management

MVC shall support efficient management of decoder resources. For instance, the output timing of multiple pictures requires efficient management. Especially, the pictures whose time stamps are the same with all views shall be available at the same time or sequentially from a decoder.

3.1.15 Parallel processing

MVC shall support parallel processing of different views or segments of the multi-view video to facilitate efficient encoder and decoder implementations.

Table 1: Test data sets

Data Set	Sequences	Image Property	Camera Arrangement
MERL	Ballroom, Exit	640x480, 25fps (rectified)	8 cameras with 20cm spacing; 1D/parallel
HHI	Uli	1024x768, 25fps (non-rectified)	8 cameras with 20cm spacing; 1D/parallel convergent
KDDI	Race1	640x480, 30fps (non-rectified)	8 cameras with 20cm spacing; 1D/parallel
KDDI	Flamenco2	640x480, 30fps (non-rectified)	5 cameras with 20cm spacing; 2D/parallel (Cross)
Microsoft	Breakdancers	1024x768, 15fps (non-rectified)	8 cameras with 20cm spacing; 1D/arc
Nagoya University	Rena	640x480, 30fps (rectified)	100 cameras with 5cm spacing; 1D/parallel
	Akko&Kayo	640x480, 30fps (non-rectified)	100 cameras with 5cm horizontal and 20 cm vertical spacing; 2D array

Table 2: Coding conditions

Test Sequence	Temporal Random Access	Basis QP			
Ballroom, Exit	0.5 sec	22	27	32	37
Uli	0.5 sec	22	27	32	37
Race1	0.5 sec	22	27	32	37
Flamenco2	0.5 sec	22	27	32	37
Breakdancers	1.0 sec	22	27	32	37
Rena [16 center views]	0.5 sec	22	27	32	37
Akko&Kayo [3 vertical * 5 horizontal views]	0.5 sec	22	27	32	37

3.2 System Support Related Requirements

3.2.1 Synchronization

MVC shall support accurate temporal synchronization among the multiple views.

3.2.2 View generation

MVC should enable robust and efficient generation of virtual views or interpolated views.

3.2.3 Non-planar imaging and display systems

MVC should support efficient representation and coding methods for 3D display including integral photography and non-planar image (e.g. dome) display systems.

3.2.4 Camera parameters

MVC should support transmission of camera parameters.

4. Test Data Sets and Test Conditions

All the proposed MVC schemes are compared using the JMVM reference S/W with the following test data sets and test conditions [9]. Multi-view test sequences vary in the number of cameras/views, the arrangement of the cameras, distance between cameras, as well as properties of the images in terms of image size and frame rate.

Table 1 describes the properties of the various test data sets. Only a subset of cameras will be used from Rena and Akko&Kayo sequences since processing all 100 views would require too much effort. In order to benefit from the high spatial density, they selected 16 central views from Rena, and chose a 2D array of size 3x5 in the vertical and horizontal directions from Akko&Kayo.

Table 2 provides the coding conditions for the MVC evaluation. The Bjontegaard measure [10] shall be used for calculating average PSNR/bitrate differences between rate-distortion (RD) curves with the delta quantization parameter (QP) values for the hierarchical-B coding structure, as specified in Table 3.

Table 3: Delta QP values

Delta QP Values	
DeltaLayer0Quant	0
DeltaLayer1Quant	3
DeltaLayer2Quant	4
DeltaLayer3Quant	5
DeltaLayer4Quant	6
DeltaLayer5Quant	7

5. Joint Multi-view Video Model (JMVM)

The multi-view video coding (MVC) is currently being developed as an extension of the ITU-T Recommendation H.264 | ISO/IEC International Standard ISO/IEC 14496-10 advanced video. In addition to implementing the normative decoding process specified in the text of the Joint Draft (JD) for MVC [11], a reference software is provided to demonstrate effectiveness of non-normative encoding techniques to be used with the standard [12].

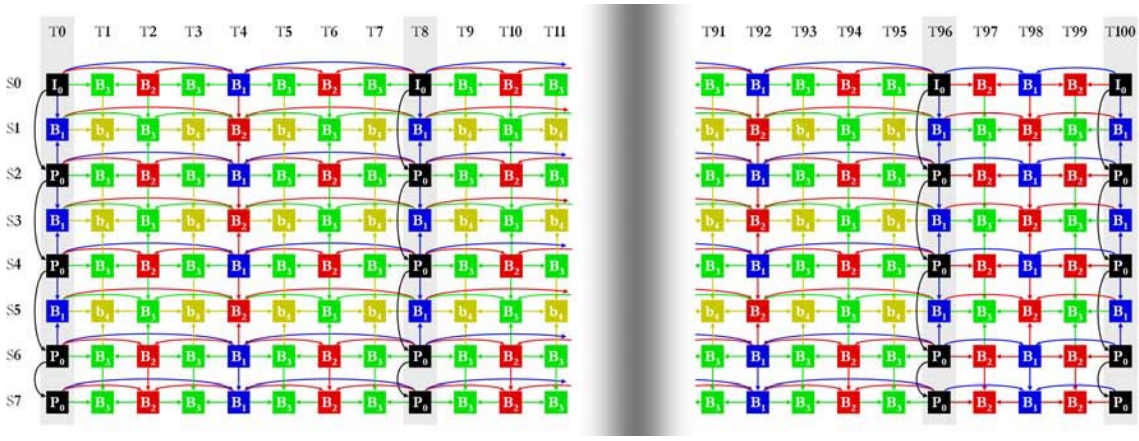

Figure 4: Inter-view-temporal prediction structure using hierarchical B pictures

As a matter of procedure, new tools, syntax or processes are firstly documented in the JMVM. Further review and software verification are conducted before the technology is adopted to the MVC standard. In observance of that procedure, decoder specifications are considered for the future version of the joint draft (JD).

Figure 4 shows the current reference structure for MVC. This scheme uses a prediction structure with hierarchical B pictures for each view. Additionally, inter-view prediction is applied to every second view: S1, S3 and S5 in Fig. 4. When the total number of views is even, the prediction structure of the last view (S7 in Fig. 4) is similar to those of even views. While B pictures in the even views do not use any inter-view references, B pictures in the last view use one inter-view reference. To allow random access, we start each GOP (S0/T0, S0/T8) with the I-frame.

Figure 4 also depicts that if the total length of the sequence does not fit an integer multiple of the GOP-length, a shortened tail GOP can be realized at the end of the sequence. In Fig. 4, the GOP-length is 8; however, we use GOP-lengths of 12 or 15 for the current test conditions.

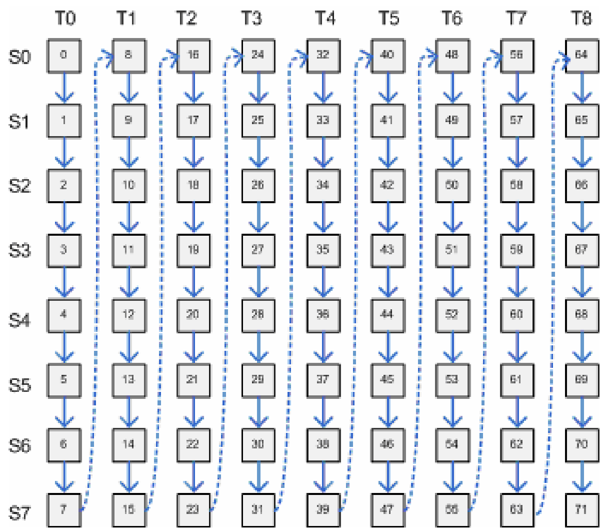

Figure 5: Coding order of multi-view videos

5.1 Prediction Structure

5.1.1 Random access

To support random access, error robustness, and synchronization, each regular group of GOPs (GGOP) contains *GOP_length* * *number_of_views* frames, e.g. 64 in Fig. 4. For accessing any frame within a GGOP, we have to decode a maximum number of frames, which basically depends on the level, the view number, and the GOP length. The maximum number of reference frames, F_{MAX}, that is necessary for B frames with the highest level *level_max* and the largest view number within a GGOP, is calculated by

$$F_{\text{MAX}} = 3 * level\_max + 2 * \lfloor (number\_of\_views - 1)/2 \rfloor \quad (1)$$

For example, in order to access B_4 (S5/T7) in Fig. 4, we need to decode the following 18 referencing frames in the hierarchical order:

- I-Frames S0/T0, S0/T8
- P-frames S2/T0, S4/T0, S6/T0, S2/T8, S4/T8, S6/T8
- B1-frames S5/T0, S5/T8, S4/T4, S6/T4
- B2-frames S5/T4, S4/T6, S6/T6
- B3-frames S5/T6, S4/T7, S6/T7.

If the *GOP_length* is 16 instead of 8, *level_max* increases by 1 and three more B-frames (B_4-frames in this example) need to be decoded. On the other hand, an increasing number of views results in an increasing number of referencing P-frames. If *number_of_views* is 15 instead of 8, F_{MAX} increases by 8 and additional P-frames to be decoded for the additional views.

5.1.2 Time-first coding order

The coding order for multi-view videos is constrained by the time-first coding. In the time-first coding, any temporal pictures are contiguous in decoding order. This can also be expressed using the following pseudo-code:

```
for ( i = 0; i < total_num_frms_per_view; i++) {
   for ( j = 0; j < total_num_views; j++) {
   jj = view_id(j);
   ii = temporal_index(i);
   process_one_frame(jj, ii);
   }
}
```

Figure 5 shows the coding order for the IPPPP prediction structure in both the temporal and the view axes.

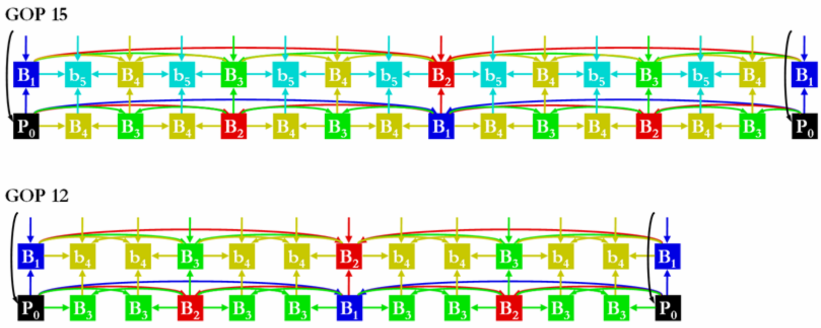

Figure 6: Basic structures for coding with GOP 15 and GOP 12

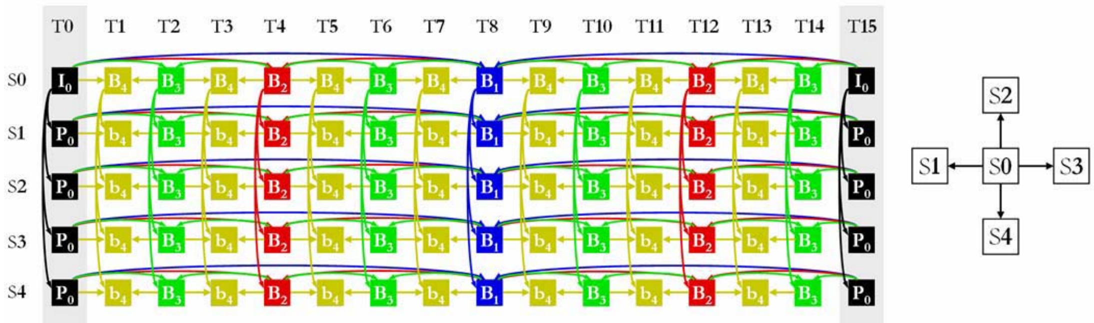

Figure 7: Modified coding scheme for Flamenco2 sequence

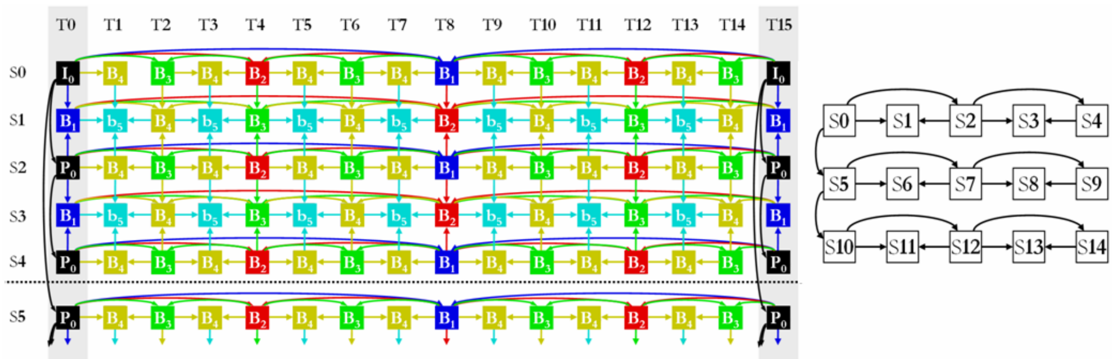

Figure 8: Modified coding scheme for Akko&Kayo sequence

5.1.3 Encoder complexity

The general complexity of the MVC reference model stems from complexity of H.264/MPEG4-AVC. The total number of frames per second is *number_of_views* times more compared to single view video.

For minimum decoded picture buffer (DPB) size is calculated by

$$DPB\ \ size = (\log_2(GOP\_length)+1)\times number\_of\_views+2 \quad (2)$$

For example, in the case of $GOP\_length = 16$ and $number\_of\_views = 8$, the DPB size would be 42. For coding delay, the MVC codec will have the same coding delay as single view video coding since time-first coding is mandated.

5.1.4 GOP structures and view prediction structure

Table 4 contains the temporal partitioning structure for each data set and Fig. 6 depicts the appropriate basic coding structure of GOPs with length 15 or 12.

The general coding structure shown in Fig. 4 can directly be applied to sequences with a 1D camera arrangement, however has to be slightly modified for the test data sets Flamenco2 and Akko&Kayo with their cross and 2D array camera setup respectively, as show in Fig. 7 and Fig. 8.

Table 4: Temporal partitioning of test data sets

Data Set	Temporal partitioning		
	Total frames	Main body	Last GOP
Ballroom	250 frames	20*GOP_12	GOP_9
Exit	250 frames	20*GOP_12	GOP_9
Uli	250 frames	20*GOP_12	GOP_9
Race1	532 frames	35*GOP_15	GOP_6
Flamenco2	1000 frames	66*GOP_15	GOP_9
Breakdancers	100 frames	6*GOP_15	GOP_9
Rena	300 frames	19*GOP_15	GOP_14
Akko&Kayo	300 frames	19*GOP_15	GOP_14

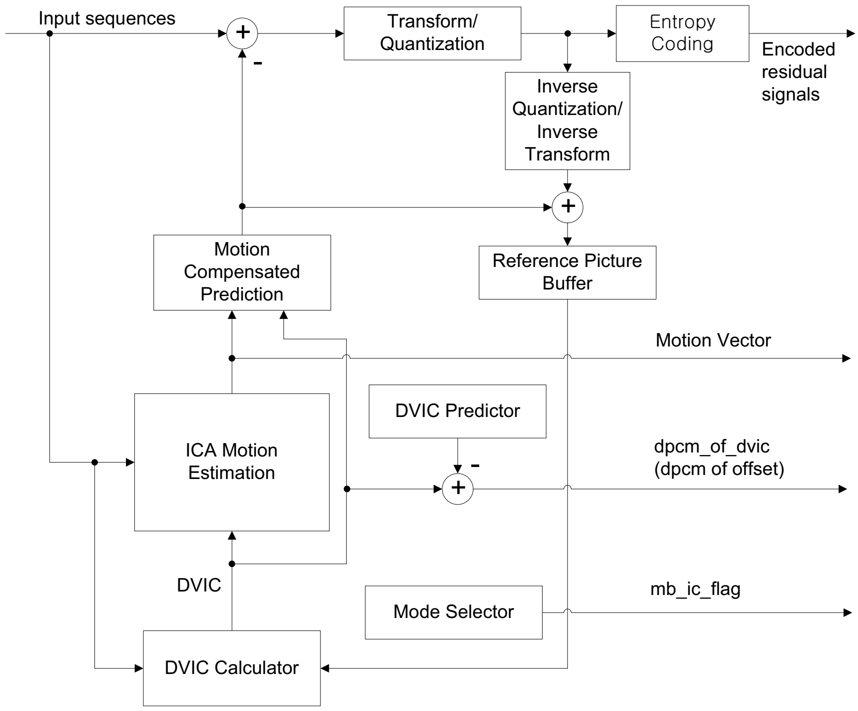

Figure 9: The encoding structure of the MB-based ICA ME/MC

5.2 Illumination Compensation

In order to compensate an illumination change between the adjacent views, the illumination change-adaptive motion compensation (ICA MC) is enabled for several macroblocks (MB) modes in H.264/MPEG-4 AVC: Inter 16×16 mode, Direct 16×16 mode (include B_Skip), and P_Skip mode. Since the difference value of illumination change (DVIC) of the current MB highly correlated with DVIC of the neighboring MB using ICA MC, the DVIC of the current block is predicted and only the offset between the current DVIC and its predicted value is coded.

Figure 9 shows the encoding structure of the illumination compensation scheme. At the encoder side, ICA motion estimation is needed for Inter 16×16 mode. Assume that the current frame is denoted by *f(i, j)* with spatial coordinates *(i, j)*, and the reference frame is *r(i, j)*. The conventional SAD calculation for the motion estimation of $S \times T$ blocks, in our case 16×16, is performed as follows:

$$SAD(x,y) = \sum_{i=m}^{m+S-1} \sum_{j=n}^{n+T-1} \left| f(i,j) - r(i+x, j+y) \right|. \quad (3)$$

where *(x, y)* represents a candidate motion vector, MV, and *(m, n)* represents a position of the current block. In order to compensate the illumination change, MR_SAD (Mean-Removed SAD) is defined as follows:

$$M_{cur} = \frac{1}{S \times T} \sum_{i=m}^{m+S-1} \sum_{j=n}^{n+T-1} f(i,j)$$
$$M_{ref}(p,q) = \frac{1}{S \times T} \sum_{i=p}^{p+S-1} \sum_{j=q}^{q+T-1} r(i,j) \quad (4)$$

$$MR\_SAD(x,y) = \sum_{i=m}^{m+S-1} \sum_{j=n}^{n+T-1} \left| \{f(i,j) - M_{cur}\} - \{r(i+x, j+y) - M_{ref}(m+x, n+y)\} \right| \quad (5)$$

where M_{cur} and $M_{ref}(p, q)$ are the average pixel values of the current block and reference block, respectively and *(p, q)* represents a position of the reference block. The final MV together with its associated DVIC (= $Round\{M_{cur} - M_{ref}(p, q)\}$) are obtained through the ICA ME based on minimum MR_SAD.

After the ICA ME is performed, an illumination compensated residual signal can be obtained as follows:

$$\begin{aligned} MR\_R(i,j) &= \{f(i,j) - M_{cur}\} - \{r(i+x', j+y') - M_{ref}(m+x', n+y')\} \\ &= \{f(i,j) - r(i+x', j+y')\} - \{M_{cur} - M_{ref}(m+x', n+y')\} \\ &= \{f(i,j) - r(i+x', j+y')\} - DVIC \end{aligned} \quad (6)$$

where *MR_R(i, j)* represents an illumination compensated residual signal and *(x', y')* represents a determined motion vector.

Since the motion vectors are derived for B_Skip/ Direct_16x16 modes, there is no ICA ME involved. For these modes, DVIC is calculated using the reference block pointed by the derived motion vector. The remaining part of the process is the same as the case of Inter 16×16 mode. For the P_Skip mode, the DVIC is derived from the two neighboring blocks as specified in the corresponding decoding description.

Recently, adaptive deblocking filtering schemes are proposed to reduce the blocking artifacts cased by the block-based illumination compensation. These schemes control the deblocking strength by considering the illumination compensation.

5.3 Other Technical Issues

5.3.1 View-temporal prediction structure

MVC researchers have studied efficient view-temporal prediction structure considering coding gain, coding delay, random access, and so on. Contrary to the single view video, the multi-view video has not only temporal correlation but also is inter-view correlation between the adjacent views. Three main coding structures were studied and experimental results of multi-view video coding have been reported [13]. The most straightforward method is to encode the multiple video sequences separately. In this method, only temporal correlation within one view is used. Another method is to utilize inter-view correlation only. In this case, images of one view are predicted from their neighboring view images. The third method utilizes both temporal and inter-view correlation. From the experiments, the third method shows the best results among the three methods.

5.3.2 View interpolation prediction

The view interpolation prediction scheme is being studied to improve the coding efficiency of MVC. This scheme generates the intermediate image in center view using the images in neighboring views, and then utilizes the synthesized intermediate image as an additional reference frame for the encoding of the images in the center view [13]. There are two main methods which is based on decoder side disparity estimation and based on computing depth at the encoder and transmitting this to the decoder. Both approaches, however, require a number of subparts. The former approach needs to rectify the decoded reference images, to generate disparity map from reference images by using stereo match, to synthesize the intermediate image from the disparity map. The latter approach needs to compute depth map which yields good compression performance, to synthesize the intermediate image from the depth map, and to transmit the calculated depth map to decoder side.

5.3.3 Motion/disparity vector coding

The motion vector generated from the inter-view prediction is called as the disparity vector [13]. In MVC, since the motion/disparity vectors highly correlated each other, efficient motion/disparity vector coding scheme efficiently can reduces bits for representing motion/ disparity vectors.

6. CONCLUSION

The multi-view video includes multi-viewpoint video sequences captured by several cameras at the same time, but different locations. Multi-view video can offer arbitrary viewpoints of dynamic scenes and thus allow more realistic video. Currently, many researchers are trying to compress multi-view video efficiently, and MPEG and JVT are leading the standardization of multi-view video coding (MVC). In this paper, we introduced the technical features of the multi-view video coding (MVC) and described its applications, requirements, and the reference software model for MVC.

Acknowledgements

This work was supported in part by the Ministry of Information and Communication (MIC) through the Realistic Broadcasting Research Center (RBRC) at Gwangju Institute of Science and Technology (GIST), and in part by the Ministry of Education (MOE) through the Brain Korea 21 (BK21) project.

References

[1] A. Smolic and P. Kauff, “Interactive 3D Video Representation and Coding Technologies,” Proceedings of the IEEE, Special Issue on Advances in Video Coding and Delivery. Vol. 93. pp. 99-110, 2005

[2] A. Smolic, K. Mueller, T. Rein, P. Eisert and T. Wiegand, “Free Viewpoint Video Extraction, Representation, Coding, and Rendering,” Proceeding of IEEE International Conference on Image Processing. Vol. 5. pp. 3287-3290, 2004

[3] R.S. Wand and Y. Wang, “Multiview Video Sequence Analysis, Compression, and Virtual Viewpoint Synthesis,” IEEE Transactions on Circuit and System for Video Technology, Vol. 10. pp. 397-410, 2000

[4] A. Redert, M. Beeck, C. Fehn, W. IJsselsteijn, M. Pollefeys, L. Gool, E. Ofek, I. Sexton, and P. Surman, “Advanced three-dimensional television system technologies,” Proc. 3D Data Processing Visualization and Transmission, 2002

[5] J.R. Ohm, “Stereo/Multiview Encoding Using the MPEG Family of Standards,” In Proceedings of Electronic Imaging '99, San Diego, USA, Jan. 1999

[6] J.R. Ohm, and K. Müller, “Core Experiments on Multiview Objects,” M3178, Feb. 1998

[7] ISO/IEC JTC1/SC29/WG11, “MPEG-4 Overview,” N4030, March 2001

[8] ISO/IEC JTC1/SC29/WG11, “Requirements on Multi-view Video Coding v.7,” N8218, July 2006

[9] ISO/IEC MPEG & ITU-T VCEG, “Common Test Condition for Multiview Video Coding,” JVT-U211, Oct. 2006

[10] ITU-T SG16/Q.6 VCEG, “Calculation of average PSNR differences between RD-Curves,” VCEG-M33, March 2001

[11] ISO/IEC MPEG & ITU-T VCEG, “Joint Draft 1.0 on Multiview Video Coding,” JVT-U209, Nov. 2006

[12] ISO/IEC MPEG & ITU-T VCEG, “Joint Multiview Video Model (JMVM) 3.0,” JVT-V207, Jan. 2007

[13] ISO/IEC JTC1/SC29/WG11, “Description of Core Experiments in MVC,” N8019, April 2006