A Modified Multiview Video Streaming System Using 3-Tier Architecture

Mohamed M. FOUAD, Hussein A. ALY

Department of Computer Engineering, Military Technical College, Ismael Elfangary St., Cairo, Egypt

mmafoad@sce.carleton.ca, haly@ieee.org.

DOI: 10.15598/aeee.v14i2.1597

Abstract. *In this paper, we present a modified inter-view prediction Multiview Video Coding (MVC) scheme from the perspective of viewer's interactivity. When a viewer requests some view(s), our scheme leads to lower transmission bit-rate. We develop an interactive multiview video streaming system exploiting that modified MVC scheme. Conventional interactive multiview video systems require high bandwidth due to redundant data being transferred. With real data test sequences, clear improvements are shown using the proposed interactive multiview video system compared to competing ones in terms of the average transmission bit-rate and storage size of the decoded (i.e., transferred) data with comparable rate-distortion.*

Keywords

Interactive streaming, multiview video coding, random access, transmission bit-rate, video compression.

1. Introduction

Multiview video consists of video sequences of the same scene captured time-synchronously by multiple closely spaced cameras from different observation viewpoints [1]. Multiview Video Coding (MVC) [2] has been used to encode the multiview video signals using various proposed schemes including both temporal and inter-view predicted frames (i.e., frames are predicted not only from the temporally neighboring frames, but also from the corresponding frames in adjacent views). MVC typically focuses on increasing the Rate-Distortion (RD) performance for the compressed frames of all views as shown in [3], [4]. Since users do not need all of the views at the same instance, transmitting the whole set of frames leads to consuming bandwidth resources. Nevertheless, decoding the compressed multiview video at the user side requires high computational cost and storage space.

An Interactive Multiview Video Streaming (IMVS) system [5] provides the aforementioned multiview video service efficiently and flexibly, to enable a viewer to freely interact with the multiview video data. The IMVS system has the advantage of reducing the bandwidth usage, since only the requested subset of the multiview video data is transmitted. However, the primary challenge in an IMVS system is to design a structure to encode the multiview video data with a reasonable compression efficiency [6] (i.e., having the transmission bit-rate reduced), having the RD performance increased, and having the storage size of the encoded multiview video data reduced.

Readers are referred to [7], [8], [9], [10], [11] for more details on IMVS systems. In [7], the IMVS system encodes the multiview video data with a simulcast mode. In such a mode, each view is encoded and transmitted independently, and each client receives as many needed views according to the channel bandwidth. Although such an IMVS system increases the interactivity between the user and the underlying requested view(s), redundant data is transferred at the expense of the quality of the transferred video for limited channel bandwidth.

In [8], a client-driven multiview video streaming system is presented to allow a user to watch 3D video interactively with significantly reduced bandwidth requirements by transmitting a small number of views selected according to the viewer's head position. That system makes use of MVC and scalable video coding concepts together to obtain improved compression efficiency. However, a base layer and enhancement layers of two selected views are additionally transmitted.

In [9], a similar IMVS system to that in [7] is designed to encode the multiview video data with a simulcast coding method, where the multiview video data is sent as two separate streams transported at separate Internet Protocol (IP) channels. However, simulcast encoded video still contains a large amount of inter-view redundant data and needs to by synchronized with inter-view switching. If two views need to be retrieved from the currently received view to another, those two views may have different end-to-end delays. Such discontinuity would negatively impact the viewing experience of end users.

In [10], an IMVS system uses successive view motion model that discriminates all frames into potential and redundant ones to be encoded and transmitted to the client. However, the performance of this system depends on Kalman filter-based predictor. If there are no prediction errors, high-quality streams are displayed. However, the predictor is not fully perfect. So, if the prediction is not fully correct, only the base layer (low-quality) is displayed and it brings poor user experience.

In [11], an encoding structure is presented to enable each view to be transmitted over a multicast group formed by clients requesting the same view. An optimal rate allocation algorithm is proposed to deliver the views selected by a client according to the network conditions. However, the decoding complexity should be maintained at a low level for a IMVS decoder due to various processing capability of terminal devices used by different interactive clients.

The IMVS system has the advantage of using a reduced bandwidth since only the requested data subset is transmitted. The primary challenge in an IMVS system is to design a structure to encode the multiview video data with a good compression efficiency, so that the transmission bit-rate is appropriately traded off with the storage size.

In [12], an MVC scheme that encodes the requested multiview video subset data, is presented. In that scheme, the inter-view prediction is performed only for the key frames to provide P-frames for both even and odd camera views. Whereas, the non-key frames of each Group of Pictures (GoP) are predicted with hierarchical B-frames in the temporal direction. In this paper, we extend the work to embed it as a first step of a proposed interactive multiview video streaming system, with a 3-tier architecture inspired from [13] (i.e., client, application server, and database server). The proposed IMVS system is compared to the state-of-the-art IMVS systems in terms of transmission bit-rate (in kb·s^{-1}) and pre-encoded data storage size (in kByte).

The rest of the paper is organized as follows. The proposed IMVS system including the MVC scheme used is described in Section 2. Implementation setup, data set sequences used and experimental results are shown in Section 3. Finally, conclusions are given in Section 4.

2. The Proposed IMVS System

A typical IMVS system consists of 5 successive steps: capture, encode, store, transmit and decode. First, the multiview video data is encoded using an encoding scheme. Then, the encoded multiview video data is submitted to a central server, called application server, in order to be stored in a video database that is available at the MVC database server.

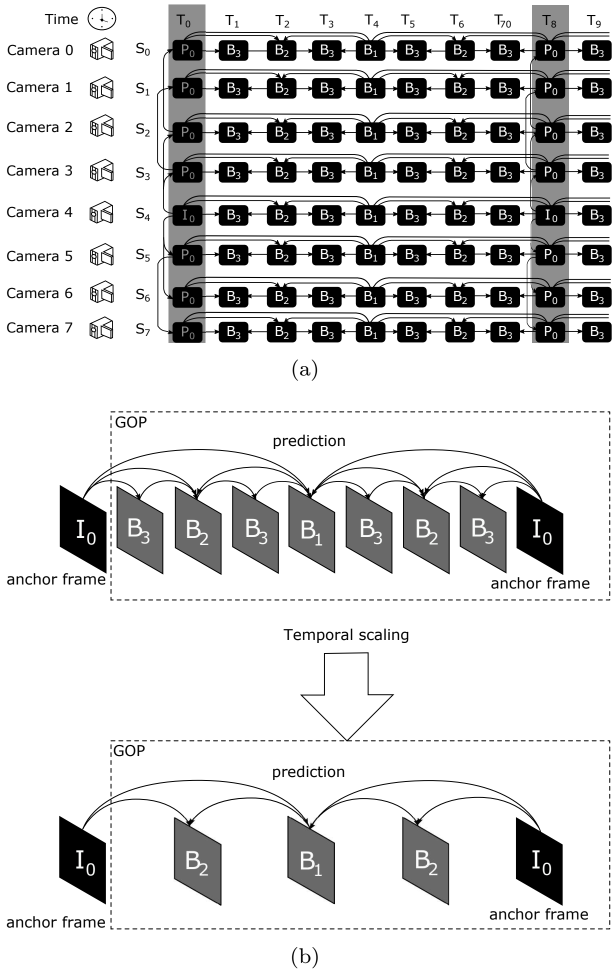
The application server only needs to prepare and transmit video stream to each client once its request has been received. The video stream is then prepared by splitting the requested multiview video subset data from the whole multiview video set. The database management system at the MVC database server fetches the prepared video stream to be submitted to the application server that returns it back to the client(s). The application server can also reduce the resolution of the video stream to adapt to the available transmission bandwidth. The resolution reduction can be obtained by decreasing the number of video frames at the time domain. Finally, at the client side, there is a standard video decoder that decodes the retrieved multiview video subset data.

The proposed IMVS system is based on 3-tier architecture: MVC encoding scheme, application server and MVC database server that are shown in Section 2.1. , Section 2.2. and Section 2.3. , respectively. The cost of splitting views and that of random access are presented in Section 2.4. and Section 2.5. respectively.

2.1. The MVC Scheme Used

This subsection shows the MVC scheme used in the proposed IMVS system that encodes the retrieved multiview video subset data.

The captured sequence is encoded by an MVC encoder that generates one merged stream. The generated bit-stream is submitted to the application server to be stored at the MVC database server. Figure 1(a) shows an example of the prediction structure of the proposed MVC scheme [12], with number of views, N, set to 8 and GoP length, M, set to 8. Setting the base view to S_4, the inter-view prediction is performed only for the key frames at T_0 and T_8 to provide P-frames for even camera views (S_2, S_0 and S_6) as well as odd camera views (S_3, S_1, S_5 and S_7). Whereas, the non-key frames of each GoP are predicted with hierarchical B-frames in the temporal direction as shown in [14]. Temporal scaling is shown in Fig. 1(b) can be applied to any multiview with more than two views.



(a)

(b)

Fig. 1: (a) The prediction structure of the proposed multiview video coding scheme, (b) The proposed temporal scaling.

2.2. The Application Server

In this subsection, the application server role is presented. The 3-tier architecture of the proposed IMVS system is shown in Fig. 2. The client selects a multiview video subset data stored in the MVC database to be decoded and displayed. The selection process acts as a request from the client to the application server. The control module at the application server, receives and schedules the clients' requests, then asks the MVC database server to retrieve the requested view(s) from the MVC database to be transferred to the client. The scheduling process is performed according to the requested view(s), client's code, and available transmission bandwidth. If there are more than one request for the same view, the control module transmits that view over a multicast group formed by the clients requesting that view. The client can randomly switch between frames in both temporal and view-wise directions. The cost of such a random access will be shown in Section. 2.5.

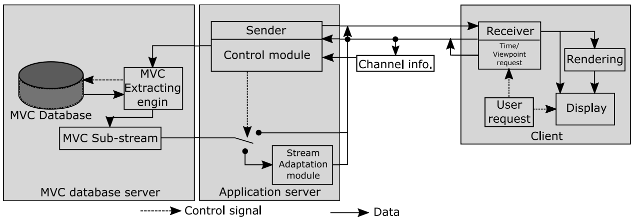


Fig. 2: The 3-tier architecture of the proposed IMVS system.

The control module checks for the available transmission bandwidth. In case of insufficient bandwidth, the requested view(s) will be accumulated at the application server. Thus, a stream delay will occur yielding buffer overflow. In such a delay case, the control module passes the video stream through the stream adaptation module. This module reduces the video stream resolution using temporal scaling in the time domain by decreasing the number of video frames within each GoP. Figure 1(b) shows the proposed temporal scaling at the hierarchical B-frames. It can be shown that the B-frames with symbol "B_3" are not used as reference frames to encode others. Thus, those frames can be discarded to reduce the number of frames within one GoP before transmission, in order to adapt to the available transmission bandwidth.

2.3. The MVC Database Server

This subsection shows the role of the MVC database server in splitting the multiview video subset data in response to the client's requests.

The video database typically provides video preprocessing for content representation and indexing, storage management for video, and continuous video streaming [15], [16]. The MVC database has the ability to split a requested view from the whole set of views to be transmitted to the client. The MVC extraction engine retrieves the requested view from the MVC database server by splitting that view to its references from the whole set of views. The output of the MVC extraction engine forms a MVC sub-stream to be submitted to the application server, before it is transmitted to the client. The cost of the view splitting step is discussed in the following subsection.

2.4. Cost of Splitting Views

As shown in Section 2.1. , the MVC prediction structure consists of one base view, S_b, and multiple enhanced views, S_e. The S_b is normally coded by single-view coding, and acts as a reference frame to encode other S_e frames. For some view, S_n, The splitting process is obtained by extracting its GoP series from each group of GoP stream. The number of extracted frames for one GoP can be generally formulated as:

$$E(b,n,l,\alpha,\beta) = I + R(b,n,\alpha)\times P + G(l,\beta)\times B, \quad (1)$$

where $E(\cdot)$ denotes the cost function for extracting frames, b denotes the base view number, n denotes the view number to be encoded, α denotes the style of inter-view prediction at key frames; $\alpha \in \{1,$ 2: 1 for standard style (i.e., is referred to as HBP), and 2 for sequential style$\}$, β denotes the number of reference views for non-key frames; $\beta \in \{0,\ 1,\ 2\}$, $R(\cdot)$ denotes a function to determine the number of key frames in an inter-view prediction as of S_b through S_n, $G(\cdot)$ denotes a function to determine the number of non-key frames in scheme related with S_n and l denotes the number of non-key frames in that GoP. The function $R(b,n,\alpha)$ in Eq. (2) can be cast as:

$$R(b,n,\alpha) = \begin{cases} \lceil |(b-n)|/2 \rceil, & \alpha = 1, \\ |(b-n)|, & \alpha = 2, \end{cases} \quad (2)$$

where $b,\ n \in\ \{0,1,2,\ldots,N-1\}$ and N denotes the total number of the views. As well, the function $G(l,\beta)$ in Eq. (1) can be written as:

$$G(l,\beta) = \begin{cases} l, & \beta = 0, \\ 2\times l, & \beta = 1, \\ 2+3\times l, & \beta = 2. \end{cases} \quad (3)$$

Generally, the cost, $Cost_E$, of the extracted frames for splitting all GoPs can be cast as:

$$Cost_E = \sum_{i=0}^{N-1} E(b,i,\alpha,\beta). \quad (4)$$

To improve the view extraction performance, the $Cost_E$, in Eq. (4), of each GoP has to be minimized. Therefore, the $Cost_E$ in Eq. (7) can be reformulated for a given encoding scheme, τ.

$$Cost_E(\tau) = \arg \min_{S_b,\alpha,\beta \in\ \{\tau\}} \sum_{i=0}^{N-1} E(b,i,l,\alpha,\beta), \quad (5)$$

where each τ has its own parameters S_b, α and β.

To solve the minimization problem in Eq. (5), we should better choose an encoding scheme to use in the proposed IMVS system. This choice step can be obtained by determining the $Cost_E$ for all candidate encoding schemes, considering that the best scheme yields the lowest $Cost_E$ value.

2.5. Cost of Random Access

It is worth noting that the random accessibility is the first step in interactivity. The user can access any single frame in either temporal or view-wise directions when watching a multiview video program [17]. Random Access (RA) can be defined as the cost of accessing any frame in one video sequence. Therefor, RA can be considered as an evaluation performance metric for a candidate prediction structure of an encoding scheme. The RA performance is measured by the number of frames that are needed to decode a specific frame in one GoP. In turn, the best encoding scheme should yield a minimum Accumulative Sum of the Reference Frames (ASRF) that can be formulated as in Eq. (6).

Where $A(\cdot)$ denotes the ASRF, b denotes the base view number; $b \in\ \{0,1,2,\ldots,N-1\}$, n is the randomly selected view number; $n \in\ \{0,1,2,\ldots,N-1\}$, $F_{n,t}$ denotes the frame at view S_n and time t, l denotes the number of non-key frames in GoP, and α and β are as defined in Eq. (1). The function $P(F_{n,t},\beta)$ determines the number of reference frames for the frame $F_{n,t}$ and can be formulated as in Eq. (7).

Where $\Lambda \equiv \{1,2,\ldots,l\}$ and ϑ denotes the level of non-key frame $F_{n,t}$ in an encoding scheme. The function $H(b,n,\alpha)$ determines the number of frames in the inter-view prediction and defined as in Eq. (8).

As well, the function $D(F_{n,l})$ determines a constant value according to the location of the frame in an encoding scheme and can be written as:

$$D(F_{n,t}) = \begin{cases} 0, & n = b,\ t \in \{1,2,\ \ldots,l\}, \\ 1, & n \neq\ b,\ t \in \{1,2,\ \ldots,l\}, \\ \frac{1}{2}, & t \in \{0,l+1\}. \end{cases} \quad (9)$$

For instance, in the proposed encoding scheme, shown in Fig. 1(a), the postscript of I-, P-, and B-frames denotes the level ϑ. For certain S_b, the cost of random access, $Cost_R$ (in frames), can be determined as:

$$Cost_R = \sum_{j=0}^{N-1} A(b,j,\alpha,\beta). \quad (10)$$

To improve the random accessibility performance, the $Cost_R$, in Eq. (10), of each GoP has to be minimized. Therefore, the $Cost_R$ in Eq. (10) can be reformulated for a given encoding scheme, τ, as:

$$Cost_R(\tau) = \arg \min_{S_b,\alpha \in\ \{\tau\}} \sum_{j=0}^{N-1} A(b,j,\alpha,\beta), \quad (11)$$

where each τ has its own parameters S_b, α, and β.

To solve the minimization problem in Eq. (11), we should better choose an encoding scheme to use in the proposed IMVS system. This choice step can be obtained by determining the $Cost_R$ for all candidate encoding schemes, considering that the best scheme yields the lowest $Cost_R$ value.

$$A(b,n,\alpha,\beta) = \sum_{t=0}^{l+1} \left\{ P(F_{n,t},\beta) + 2 \sum_{k\in \{H(b,n,\alpha)\}} D(F_{n,t}) \right\}, \tag{6}$$

$$P(F_{n,t},\beta) = \begin{cases} \vartheta + 1, & \beta = 0,\ t \in \Lambda, \\ 2(\vartheta + 2) - 1, & \beta = 1,\ t \in \Lambda, \\ (\vartheta \times 2) - (\vartheta - 2), & \beta = 2,\ t \in \Lambda, \\ 0, & t \in \{0, l+1\}, \end{cases} \tag{7}$$

$$H(b,n,\alpha) = \begin{cases} \{b,\ b+2,\ b+4,\dots,n\}, & n > b,\ \alpha = 1, \\ \{b,\ b-2,\ b-4,\dots,n\}, & n < b,\ \alpha = 1, \\ \{b,\ b+1,\ b+2,\dots,n\}, & n > b,\ \alpha = 2, \\ \{b,\ b-1,\ b-2,\dots,n\}, & n < b,\ \alpha = 2, \\ \{b\}, & n = b. \end{cases} \tag{8}$$

3. Experiments & Results

In this section, the data sequences used are described in Section 3.1. The implementation setup of all experiments is given in Section 3.2. Finally, the results are shown and discussed in Section 3.3.

3.1. Data Sets Description

The data set used in the experiments includes four standard video sequences [18], [19]. Their characteristics are provided in Tab. 1. The first sequence, **Ballroom** [18], shows a dynamic scene containing fast motion of the dancers and many overlapping objects. The second sequence, **Exit** [18], represents a static scene with few persons slowly moving from right to the middle of the scene. The third sequence, **Vassar** [18], has been captured in an ambient day light and contains no discernable motion blur on the boundaries of the moving objects. The fourth video sequence, **Breakdancers** [19], represents a scene captured by cameras placed on an arc-shaped alignment around the static scene, with few breakdancers.

3.2. Implementation Setup

Our implementation runs on a personal computer with 2.4 GHz Core i3 and 2GB of RAM. For the application server, we installed *Live555* media server for video transmission software [20].

In this paper, we use the joint multiview video coding software (v.8.5) [21] for encoding the data sets to extract the MVC sub-stream at the MVC extraction engine. The quantization parameter is set to 24, 28, 32, and 36. The search mode is set to fast search with search window set to 96 × 96 pixels. The length of a GoP: M is set to 12 for the **Ballroom**, **Exit**, and **Vassar** video sequences. Whereas, the length of a GoP: M is set to 15 for the **Breakdancers** video sequence .

3.3. Results of the MVC Scheme Used

The MVC scheme used [12], shown in 2.1. , is compared to:

- the simulcast scheme [4] (i.e., referred to as Simulcast),
- the encoding multiview video structure [4] (i.e., referred to as KS-IPP),
- the MVC standard scheme [22] (i.e., referred to as MVC-HBP),
- the MVC encoding scheme of [23] (i.e., referred to as YANG).

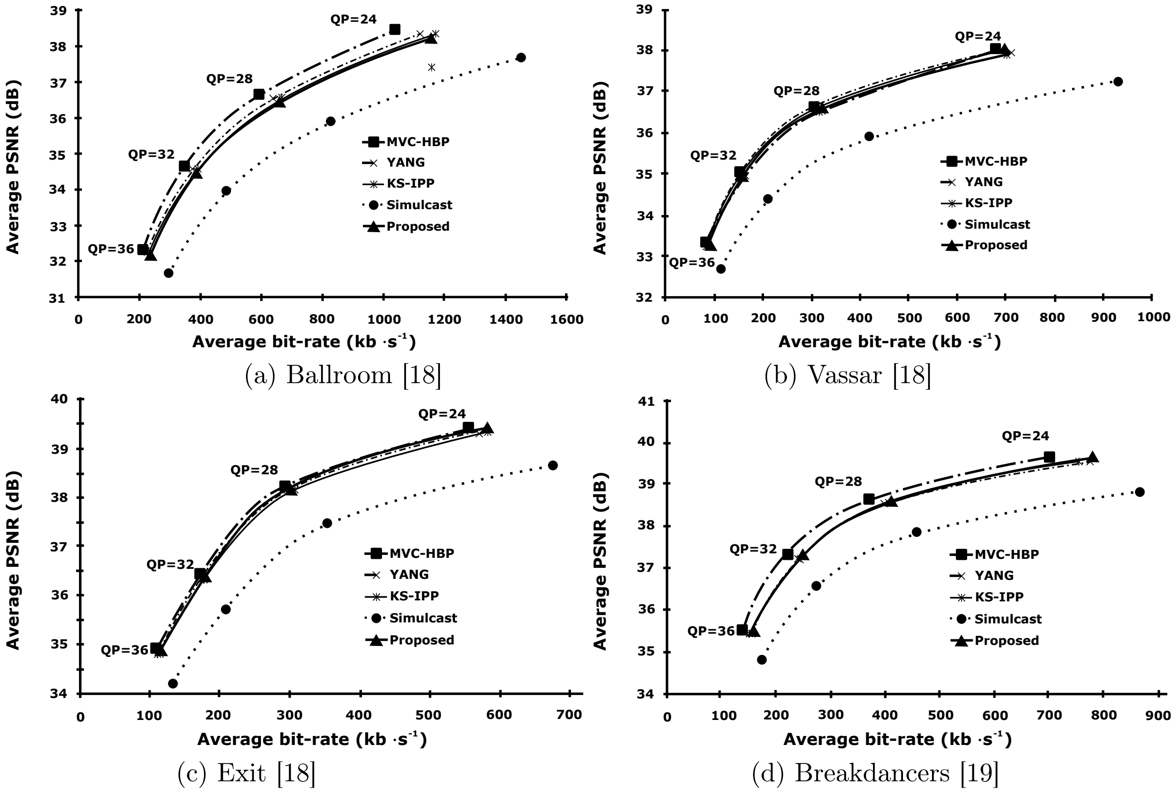
The performance of competing schemes is evaluated by three metrics:

- the RD performance (in dB·(kb^{-1}·s^{-1}) at the basis of the higher the better,
- the cost of splitting views (in frames) at the basis of the lower the better,
- the cost of random access (in frames) at the basis of the lower the better.

The RD performance, measured in dB·kb^{-1}·s^{-1}, describes the trade-off between the video quality and the bit-rate of the video stream. The RD performance is at the basis of the higher the RD value, the better

Tab. 1: Description of the test video sequences [18], [19]

Sequences	Object Motion	Resolution	Format	Camera Arrangement	File size in 8 views	Bit-rate (kb·s^{-1})
Ballroom Vassar Exit	Medium Low High	640×480 rectified, 25 fps	4:2:0	8 cameras 1D/parallel 20 cm inter-spacing	878 MB 10 s	719 257.6
Breakdancers	High	1024×768 15 fps	4:2:0	8 cameras 1D/arc 20 cm inter-spacing	900 MB 6.7 s	1 100 417.9



(a) Ballroom [18]

(b) Vassar [18]

(c) Exit [18]

(d) Breakdancers [19]

Fig. 3: Rate-distortion performance of competing encoding schemes: *MVC-HBP* [22], *YANG* [23], *KS-IPP* [4], *Simulcast* [4], and *Proposed* [12], for different sequences with different quantization parameters (QP) using standard video sequences.

the MVC encoding scheme. Figure 3 shows the RD performance using competing MVC schemes applied to the data sequences shown in Section 3.1. at different quantization parameters. It can be shown that the MVC scheme used provides comparable RD performance compared to the KS-IPP [4], MVC-HBP [22] and YANG [23] schemes. Whereas, the MVC scheme used surpasses the Simulcast scheme [4] by an average improvement of 19 % in terms of RD performance.

Table 2 shows the cost of splitting views, $Cost_E$, for competing MVC schemes. The MVC scheme used outperforms the MVC-HBP [22], KS-IPP [4], and YANG [23] schemes by an average reduction of 44.6 %, 14.2 % and 3 %, respectively, in terms of the cost of splitting views.

Tab. 2: Cost of splitting views, $Cost_E$, using competing MVC schemes with different groups of GoP. The lower, the better.

Group of GoP size	$Cost_E$ (in frames)			
	MVC-HBP [22]	KS-IPP [4]	YANG [23]	Prop.
8×8	132	92	77	**74**
8×12	192	124	109	**106**
8×15	237	142	133	**130**

Table 3 shows the cost of random access, $Cost_R$, that can be determined by Eq. (11) using all competing MVC schemes. It can be shown that the MVC scheme used outperforms the MVC-HBP [22], KS-IPP [4], and the YANG [23] schemes by an average reduction of 42.9 %, 43.2 % and 1.1 % respectively, in terms of the cost of random access.

Tab. 3: Cost of random access, $Cost_R$, using competing MVC schemes with different groups of GoP. The lower, the better.

Group of GoP size	$Cost_R$ **(in frames)**			
	MVC-HBP [22]	**KS-IPP [4]**	**YANG [23]**	**Prop.**
8×8	615	640	358	**352**
8×12	979	992	566	**560**
8×15	1357	1312	778	**772**

3.4. Results of the Proposed IMVS System

The proposed IMVS system, referred to as *Proposed IMVS*, is compared to:

- the multiview video coding system [4] (i.e., referred to as MVC system),
- the real-time transmission system of high-resolution multiview stereo video over IP networks [9] (i.e., referred to as Multiview over IP system),
- the client-driven selective streaming system for multiview video transmission [11] (i.e., referred to as Client-driven system).

The performance of competing systems is evaluated by three metrics:

- the transmission bit-rate (in kb·s^{-1}),
- the pre-encoded data storage size (in kByte),
- the ratio between transmission bit-rate and storage size (in (kb·s^{-1})·kByte^{-1}).

The transmission bit-rate metric is measured in Kbps and comes at the basis of the lower the better. Table 4 shows that the proposed IMVS system outperforms the MVC [4], the Multiview over IP [9], and the Client-driven [11] systems by an average improvement of 81.8 %, 63.5 % and 42.4 %, respectively, in terms of transmission bit-rate (in kb·s^{-1}). This improvement can be analyzed as follows. The proposed IMVS system as well as the Client-driven system transmit only the requested view(s) to the client. Whereas, the MVC system transmits the whole set of views to the client. While, the Multiview over IP system [9] transmits the whole set of views into two separate streams to the client. The storage size, in KBytes, of the pre-encoded multiview video subset data is an important factor that impacts the IMVS system performance. Therefore, that factor comes at the basis of the lower the better. Table 4 shows that the proposed IMVS system outperforms the Multiview over IP system [9] by an average reduction of 18 %, and provides a negligible increase in the storage size compared to the MVC [4] and Client-driven [11] systems. In terms of the ratio between transmission bit-rate and storage size (in (kb·s^{-1})·kByte^{-1}), Tab. 4 shows that the proposed IMVS system outperforms the MVC [4], the Multiview over IP [9] and the Client-driven [11] systems by an average improvement of 79 %, 68 % and 39 %, respectively.

Tab. 4: Results of competing IMVS systems to encode standard video sequences at different quantization parameters (24, 28, 32, and 36) using i) transmission bit-rate (kb·s^{-1}), ii) storage size (kByte), and iii) transmission bit-rate/storage size ((kb·s^{-1})·kByte^{-1}).

Approach	**Metric**	**Video sequences**			
		Ballroom	**Vassar**	**Exit**	**break**
MVC [4]	(i)	4360	2431.6	2253	2865.7
	(ii)	**5322.2**	**2968.3**	**2750.3**	**2332.1**
	(iii)	0.8192	0.8192	0.8192	1.2288
Multiview over IP [9]	(i)	3029	1688.1	1370	1791.1
	(ii)	7395	4121.2	3344.6	2915.2
	(iii)	0.4096	0.4096	0.4096	0.6144
Client -driven [11]	(i)	1512.9	1118.6	775	682
	(ii)	5909.5	3087.2	2877.3	2544.2
	(iii)	0.2560	0.3623	0.2694	0.2681
Proposed IMVS system	(i)	**702**	**612.8**	**504.7**	**533**
	(ii)	5909.2	3087.5	2877.1	2544
	(iii)	**0.1188**	**0.1985**	**0.1754**	**0.2095**

4. Conclusions

In this paper, we first presented an inter-view prediction structure of the MVC scheme. The MVC scheme surpasses the KS-IPP, MVC-HBP and YANG MVC schemes by an average reduction of 44.6 %, 14.2 % and 3 %, respectively, in terms of splitting views cost and by an average reduction of 42.9 %, 43.2 % and 1.1 % respectively, in terms of the random access cost. The presented MVC scheme provides comparable rate-distortion performance compared to the aforementioned MVC schemes and surpasses the Simulcast scheme by an average increase of 19 %.

The proposed IMVS system exploits the MVC scheme used in [12] to ultimately improve the viewer interactivity. The proposed IMVS system outperforms the MVC, Multiview over IP and Client-driven system by an average improvement of 81.8 %, 63.5 % and 42.4 %, respectively, in terms of transmission bit-rate and by an average improvement of 79 %, 68 % and 39 %, respectively in terms of the ratio between transmission bit-rate and storage size. However, the proposed IMVS system has subtle increase in the storage size compared to the MVC and Client-driven systems, though the former outperforms the Multiview over IP system by an average reduction of 18 % in the storage size.

References

[1] KUBOTA, A., A. SMOLIC, M. MAGNOR, M. TANIMOTO, T. CHEN and C. ZHANG. Multiview imaging and 3DTV. *IEEE Signal Processing Magazine*. 2007, vol. 24, iss. 6, pp. 10–21. ISSN 1053-5888. DOI: 10.1109/MSP.2007.905873.

[2] ITU/T and ISO/IEC JTC 1. *Advanced video coding for generic audiovisual services. ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC): including all versions 19*. Geneva: ITU/T, May 2003–June 2011.

[3] MERKLE, P., K. MULLER, A. SMOLIC and T. WIEGAND. Efficient compression of multiview video exploiting interview dependencies based on H.264/MPEG4-AVC. In: *IEEE International Conference on Multimedia and Expo*. Toronto: IEEE, 2006, pp. 1717–1720. ISBN 1-4244-0366-7. DOI: 10.1109/ICME.2006.262881.

[4] MERKLE, P., A. SMOLIC, K. MULLER and T. WIEGAND. Efficient prediction structures for multiview video coding. *IEEE Transactions on Circuits and Systems for Video Technology*. 2007, vol. 17, no. 11, pp. 1461–1473. ISSN 1051-8215. DOI: 10.1109/TCSVT.2007.903665.

[5] CHEUNG, G., A. ORTEGA and T. SAKAMOTO. Coding structure optimization for interactive multiview streaming in virtual world observation. In: *IEEE 10th Workshop on Multimedia Signal Processing*. Cairns: IEEE, 2008, pp. 450–455. ISBN 978-1-4244-2294-4. DOI: 10.1109/MMSP.2008.4665121.

[6] CHEUNG, G., A. ORTEGA and N. M. CHEUNG. Interactive streaming of stored multiview video using redundant frame structures. *IEEE Transactions on Image Processing*. 2011, vol. 20, no. 3, pp. 744–761. ISSN 1057-7149. DOI: 10.1109/TIP.2010.2070074.

[7] KIM, J., K. CHOI, H. LEE and J. W. KIM. Multi-view 3D video transport using application layer multicast with view switching delay constraints. In: *3DTV Conference*. Kos Island: IEEE, 2007, pp. 1–4. ISBN 978-1-4244-0722-4. DOI: 10.1109/3DTV.2007.4379478.

[8] KURUTEPE, E., M. R. CIVANLAR and A. M. TEKALP. Client-driven selective streaming of multiview video for interactive 3DTV. *IEEE Transactions on Circuits and Systems for Video Technology*. 2007, vol. 17, no. 11, pp. 1558–1565. ISSN 1051-8215. DOI: 10.1109/TCSVT.2007.903664.

[9] ZHOU, Y., C. HOU, Z. JIN, L. YANG, J. YANG and J. GUO. Real-time transmission of high-resolution multi-view stereo video over IP networks. In: *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*. Potsdam: IEEE, 2009, pp. 1–4. ISBN 978-1-4244-4317-8. DOI: 10.1109/3DTV.2009.5069657.

[10] PAN, Z., Y. IKUTA, M. BANDAI and T. WATANABE. A user dependent system for multi-view video transmission. In: *International Conference on Advanced Information Networking and Applications (AINA)*. Singapore: IEEE, 2011, pp. 732–739. ISBN 978-1-61284-313-1. DOI: 10.1109/AINA.2011.31.

[11] ZHENG, S. and Z. JUNNI. A Client-Driven Selective Streaming System for Multi-view Video Transmission. In: *Advances on Digital Television and Wireless Multimedia Communications*. Berlin: Springer, 2012, pp. 372–379. ISBN 978-3-642-34594-4. DOI: 10.1007/978-3-642-34595-1_51.

[12] HAMADAN, A. M., H. A. ALY and M. M. FOUAD. A modified interview prediction scheme of multiview video coding to improve view's interactivity. In: *8th International Conference on Computer Vision Theory and Applications*. Barcelona: Springer, 2013, pp. 35–40. ISBN 978-989-8565-47-1. DOI: 10.5220/0004198600350040.

[13] ECKERSON, W. Three Tier Client/Server Architecture: Achieving Scalability, Performance, and Efficiency in Client Server Applications. *Open Information Systems* 1995, vol. 10, iss. 1, pp. 6–10. ISSN 1874-1339.

[14] SCHVARZ, H., D. MARPE, and T. WIEGAND. Analysis of hierarchical B pictures and MCTF. In: *IEEE International Conference on Multimedia and Expo*. Toronto: IEEE, 2006, pp. 1929–1932. ISBN 1-4244-0366-7. DOI: 10.1109/ICME.2006.262934.

[15] AREF, W., A. CATLIN, A. ELMAGARMID, J. FAN, J. GUO, M. HAMMAD, I. ILYAS, M. MARZOUK, S. PARABHAKAR, A. REZGUI, S. TEOH, E. TERZI, Y. TU, A. VAKALI and X. ZHU. A distributed database server for continuous media. In: *18th International Conference on Data Engineering*. San Jose: IEEE, 2002, pp. 490–491. ISBN 0-7695-1531-2. DOI: 10.1109/ICDE.2002.994764.

[16] FIOLEK, A. and D. W. COLLINS. Video data management system archives and provides online access to NOAA deep-sea corals digital video and image data. In: *OCEANS 2008*. Quebec

City: IEEE, 2008, pp. 1–6. ISBN 978-1-4244-2619-5. DOI: 10.1109/OCEANS.2008.5151940.

[17] JTC1/SC29/WG11. *Requirements on multi-view video coding v.6.* Montreux: JTC, 2006.

[18] *mvc-testseq* [online]. Available at: `ftp://ftp.merl.com/pub/avetro/`.

[19] *Microsoft Research* [online]. Available at: `http://research.microsoft.com/enus/um/people/sbkang/3dvideodownload/`.

[20] *The Live555^{TM} Media Server* [online]. Available at: `http://www.live555.com/mediaServer/`.

[21] *JVT-AD207* [online]. 2009. Available at: `http://wftp3.itu.int/av-arch/jvt-site/200901-Geneva/JVT-AD207.zip`.

[22] VETRO, A., P. PANDIT, H. KIMATA, A. SMOLIC and Y. K. WANG. JVT-AB204 *Joint draft 9.0 Multi-view Video Coding.* Hannover: JVT, 2008.

[23] YANG, Y., Q. DAI, G. JIANG and Y. HO. Comparative interactivity analysis in multiview video coding schemes. *ETRI Journal.* 2010, vol. 32, no. 4, pp. 566–576. ISSN 1225-6463. DOI: 10.4218/etrij.10.0109.0391.

About Authors

Mohamed M. FOUAD received the B.Sc. degree (excellent with honors) in Computer Engineering, and the M.Sc. degree in Electrical Engineering from the Military Technical College (MTC) Cairo, Egypt, in 1996 and 2001, respectively. As well, he received the Ph.D. degree in Electrical and Computer Engineering from Carleton University, Ottawa, Canada in 2010. He is currently a faculty member with the Department of Computer Engineering, MTC. His research interests are in online handwritten recognition, image processing, and multi-view video coding. Dr. Fouad is an IEEE Member since 2010.

Hussein A. ALY received the B.Sc. (excellent with honors) in Computer Engineering, and the M.Sc. degree in Electrical Engineering from the MTC, in 1993 and 1997, and the Ph.D. in Electrical Engineering from the University of Ottawa in 2004. He is Associate Professor with the computer department at MTC and the chief of department (2010-2013). He was a visiting professor in the electrical engineering department at the university of Rochester (Sep. 2012- Mar. 2013). His research interests are in image sampling theory and sampling structure conversion. His current research is focused on high-quality image magnification, interpolation of color filter array data, the application of total-variation for image processing, data fusion, video steganography and embedded computer systems for video processing. Dr. Aly is an IEEE Senior Member since 2011.