Low Complexity Independent Multi-ViewVideo Coding

Hussein, Hany S.
Electronics and Communications Engineering Department, Egypt-Japan University of Science and Technology (E-JUST)

El-Khamy, Mostafa
Electrical Engineering Department, Alexandria University

Mehdipour, Farhad
E-JUST Center, Kyushu University

El-Sharkawy, Mohamed
Electronics and Communications Engineering Department, Egypt-Japan University of Science and Technology (E-JUST)

出版情報：Consumer Communications and Networking Conference (CCNC), 2013 IEEE, pp.37-42, 2013-01. Institute of Electrical and Electronics Engineers
バージョン：

Low Complexity Independent Multi-View Video Coding

Hany S. Hussein[1], Mostafa El-Khamy[2], Farhad Mehdipour[3], and Mohamed El-Sharkawy[1]
[1]Electronics and Communications Engineering Department,
Egypt-Japan University of Science and Technology (E-JUST), New Borg El-Arab 21934, Alexandria, Egypt
[2]Electrical Engineering Department, Alexandria University, Alexandria 21544, Egypt
[3] E-JUST Center, Kyushu University, Fukuoka, Japan
hany.hussein@ieee.org, m_elkhamy@ieee.org, farhad@ejust.kyushu-u.ac.jp, melshark@iupui.edu

Abstract— In 3D multi-view video coding (MVC), disparity estimation (DE) are used to exploit the correlation among different view sequences. The DE process greatly increases the computational complexity of the MVC. In this paper, a novel independent low complexity multi-view video coder (I-MVC) is introduced. In the proposed MVC, the coding complexity is shifted from the encoder side to the decoder side. Instead of disparity estimation, the proposed I-MVC deploys independent component analysis (ICA) on the video streams to remove the correlation between the view sequences. The correlated (dependent) video sequences are decomposed into uncorrelated (independent) sequences and a mixing matrix. Each independent sequence is independently encoded by the H.264/AVC video coder. Then the mixing matrix is used at decoder to jointly decode the received independent sequences. Our experimental results show that the proposed I-MVC has better coding efficiency than conventional 3D multi-view video coder. The I-MVC gives more than 21% savings in overall bit rate and reduces the MVC computational complexity by 49% with less than 0.2 dB loss in the video peak signal to noise ratio.

Index Terms—3D video, Independent component analysis (ICA), joint multi-view video model (JMVM), multi-view video coding (MVC).

I. Introduction

3D multi-view video (MVV) is quickly gaining momentum due to the advancement in 3D display technologies and multi-camera arrays. In multi-view video, the same scene is captured from different viewpoints using many cameras. MVV applies to many new technologies such 3D television (3DTV) [1], high definition TV (HDTV), and free viewpoint video (FVV) [2]. The 3D MVV creates an enhanced user experience and its potentially growing market has recently attracted a lot of research

There are some challenges for 3D multi-view video (MVV) systems to become practical and ubiquitous. Due to the large amount of captured data, multi-view video coding (MVC) must be efficient to enable efficient video storage and transmission. The basic idea employed in all related works on efficient MVC is to use both spatial and temporal redundancy for Since all cameras capture the same scene from different viewpoints, there exists significant inter-view redundancy [3]. The Joint Multi-view Video Model (JMVM), introduced by the Joint Video Team (JVT) [4], is based on the H.264/AVC hierarchical B-pictures prediction structure [5]. A general MVC prediction structure is shown in Fig. 1 [5]. Frames are not only predicted from temporal neighbors, but also from spatial neighbors in different views.

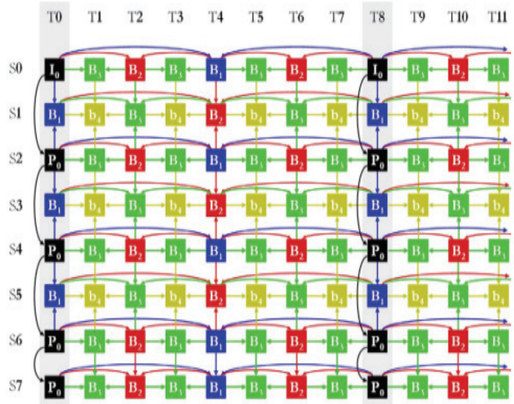

Fig. 1.The B picture hierarchical structure for MVC.

However, JMVM [4] still has several drawbacks, noticeably the high computational complexity of JMVC [6, 7] and the group of picture prediction structure (GOP-PS) configurations. Many authors addressed the computational complexity problem of JMVC. For example, the geometric arrangement of cameras can be adjusted to speed up the disparity estimation process and improve decoding performance [8, 9]. To reduce the computational complexity, the relationship between previously estimated disparity and motion vectors can be used to reduce the search range of the new motion estimation (ME) and DE processes [10]. The selection of GOP-PS has been addressed in previous research as well. For example in

[11], a correlation analysis between video sequences is used to select convenient GOP prediction structures. The temporal and spatial correlations can also be exploited to adaptively choose one out five predefined GOP prediction structures [12]. Blind calculation of efficient GOP-PS without prior knowledge of the cameras geometric arrangement can improve decoder performance as presented in [8].

However, it remains a challenge to simultaneously find the best GOP-PS and reduce the computational complexity. Existing approaches for determining the GOP prediction structure (GOP-PS) depend mainly on calculating the temporal and inter-view correlations which adds more computational complexity to the 3D video coder. In this paper, we introduce a low complexity multi-view video. The proposed coder utilizes independent component analysis [13] to decompose the correlated (dependent) video sequences into uncorrelated (independent) sequences and a mixing matrix. Thus any straightforward video coding scheme can be applied, in which each view is encoded independently. The proposed coder is thus called Independent Multi-view Video Coder (I-MVC). Using independent video coding will significantly reduce the computational complexity as it eliminates the need for the DE process. Moreover, in the I-MVC, the GOP-PS selection problem is totally solved, as the streams are independently encoded and don't require a GOP-PS.

The rest of the paper is organized as follows: in Section II we introduce a short overview on independent component analysis (ICA). Section III describes the proposed independent multi-view video coder (I-MVC) algorithm. In Section IV we discuss about experiment results, and finally Section V concludes the paper.

II. INDEPENDENT COMPONENT ANALYSIS OF MVC

Independent component analysis (ICA) [13] is an optimization method for extracting the signals or sources from multidimensional observations. Consider a number of unknown signals *S*, which have been combined in an unknown manner by the mixing matrix *A* as in (1). The objective of ICA is to recover the original signals *S* from the signals *X*. The ICA can estimate the signals *S* and the mixing matrix *A* under the assumption that the underlying sources *S* are statistically independent.

$$X = AS = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} S \qquad (1)$$

where *X* is the of the observed signals (mixed signals), *A* is the mixing matrix and *S* is vector of the original signals.

One of the fastest and most accurate ICA algorithms is the fast fixed-point algorithm for independent component analysis (FastICA) [14]. FastICA is based on the non-gaussianity maximization [14]. In the FastICA algorithm, the fourth order cumulant (kurtosis) is used as a measurement for the non-gaussianity [14]. Thus the FastICA algorithm tries to find the separation matrix A^{-1} that maximize the value of kurtosis of the resulting signals *S*.

In this paper, FastICA is used to decompose the key frames of the video streams into independent components (ICs) *S* and a mixing matrix *A*. To make sure that the samples of the ICs (*S*) are in the same video (frames) domain we project the ICs data into the frame domain as follows. The system (1) can be rewritten in the additive form [15] as:

$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11} \\ \vdots \\ a_{1n} \end{bmatrix} s_1 + \begin{bmatrix} a_{21} \\ \vdots \\ a_{2n} \end{bmatrix} s_2 + \cdots + \begin{bmatrix} a_{n1} \\ \vdots \\ a_{nn} \end{bmatrix} s_n \quad (2)$$

$$X = X_{p1} + X_{p2} + \cdots + X_{pn} \qquad (3)$$

The model (1) is a multiplicative model which can be read as: the observed vector *X* is the product of a mixing matrix *A* by a source vector *S*. While, the model (3) is an additive model which is indicating that the observed vector X is a sum of *n* one-dimensional independent vectors $X_{p1}, \ldots, X_{pn}$. It is noted that X_{pi} is a one-dimensional linear subspace spanned by the i^{th} column of the mixing matrix *A* and is equal to:

$$X_{pi} = P_i \left(\sum_{j=1}^{n} P_j\right)^{-1} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \qquad (4)$$

where P_i is the orthogonal projection onto the subspace X_{pi} and is equal to [15]:

$$P_i = \frac{A_i A_i^T}{A_i^T A_i} \qquad (5)$$

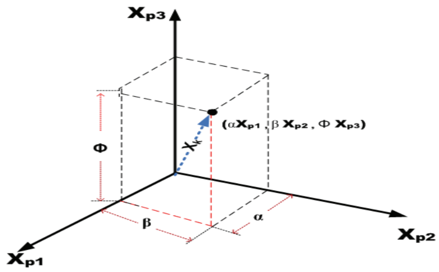

Fig. 2. Projection into pixel (frame) domain

To explain the projection into frame or video domains, assume that there are only three video streams X_1 to X_3. Thus, there will be three one-dimensional linear subspaces X_{p1}, X_{p2}, and X_{p3}. Therefore, any vector (video stream) X_k can be decomposed into three orthogonal components inside the coordinate space span of the subspaces X_{p1}, X_{p2}, and X_{p3}, as shown in Fig. 2.

The process of video decomposition and projection onto the frame (video) domain can be described as follows:

Algorithm 1: Find the new orthogonal separation matrix H.

1.1. Extract the *key frames* (first frame in the video stream) from each video stream.

1.2. Construct X:

a) Apply a 2D N-level discrete Wavelet packet transform on the luminance (Y) component of each key frame (here we used N=3) and convert the resulting approximation coefficients from a 2D to a 1D vector using the Zig-Zag scan.

b) Construct the matrix X such that each row represents the approximation coefficients of a different key frame.

1.3. Apply the FastICA algorithm on X to get the mixing matrix A.

1.4. From the mixing matrix A, find the orthogonal projectors P_i for every column in A, using (5).

1.5. Construct the new matrix M where the i^{th} row in M is picked from the projector matrix P_i.

1.6. Finally find the new orthogonal separation matrix H according to (6)

$$H = M\left(\sum_{j=1}^{n} P_j\right)^{-1} \tag{6}$$

III. INDEPENDENT MVC

In this section, we describe the proposed I-MVC algorithm and its corresponding encoder/ decoder.

A. The I-MVC encoder design.

Fig. 3 shows the flowchart of the proposed encoder .The encoder checks the frame types of the input video streams according to the intra frame period of the B picture hierarchical structure. If a frame is an intra (key) frame, the Algorithm 1 is applied to find the new orthogonal separation matrix H. Then H is used to estimate the independent components (I_v) using (7)

$$\begin{bmatrix} Iv_1 \\ \vdots \\ Iv_r \\ \vdots \\ Iv_n \end{bmatrix} = H \begin{bmatrix} X_1 \\ \vdots \\ X_r \\ \vdots \\ X_n \end{bmatrix} \tag{7}$$

Each component X_i of X refers to one complete video stream, thus we will denote the i^{th} video stream by X_i and its independent video component by I_{vi}. Finally, the H.264 video encoder is applied on each I_v component independently.

At the encoder, the intra-frames (key frames) of the input video streams are used to estimate orthogonal separation matrix H (Algorthim1). Since the intra frames of each video stream are encoded independently of any other frames within the same stream, the intra frames of the independent components (I_v) are replaced with the intra frames of the original input video stream X. This operation has two properties:

P1. It eliminates the need for sending H to the decoder hence, less communication overhead.

P2. The intra frames will be used at the decoder side to enhance the quality of the decoded video sequences as will be explained in Section III.B.

The gray color block in Fig. 3 is inserted to replace one of the independent video (I_v) with one of the input video sequences V_r which is called the reference video stream.

Algorithm 1 deploys ICA to remove the redundancies between the input video streams X. The aim of ICA is to transform the input video stream X from the pixel domain (dependent domain) into independent components I_v (residual domain). Thus, the residual ICs I_v are uncorrelated up to higher order statistics. This guarantees that any residual component I_{vi} can be encoded with a lower bit rate than the input video stream X_i. Therefore, the encoded bit rate will reduce. Moreover, the encoding computational complexity will reduce.

The encoder algorithm can be summarized as follows:

Algorithm 2: Encoder algorithm.

2.1. Input the video streams V_1..., V_r... V_n.

2.2. Choose one of the input streams as a reference video stream V_r.

2.3. Check the input frames type and start with intra frames.

2.4. If the input frames type is intra (I) frame.

i. Estimate H by using Algorithm 1.

ii. Encode the intra frames directly.

2.5. Else apply (7) to remove the redundancy between the input frames.

2.6. Encode all the independent videos (I_v) except the independent video I_{vr}, which is replaced with the reference video stream V_r.

B. The I-MVC decoder design.

The first step in the decoder is to decode the received bit streams to generate the independent components and the reference video streams V_r. However, the coefficients of the independent component (I_v) are sensitive to quantization error and the rounding process. Thus Algorithm 1 is applied on the received intra frames (P1) to estimate a new version of the orthogonal separation matrix H. The new version of H which is estimated at the decoder side will be affected by the same quantization and the rounding errors. Thus the new H matrix will improve the quality of the recovered video streams. The decoder can recover the video streams X by using the H matrix, the received independent components (I_v) and the reference video stream V_r.

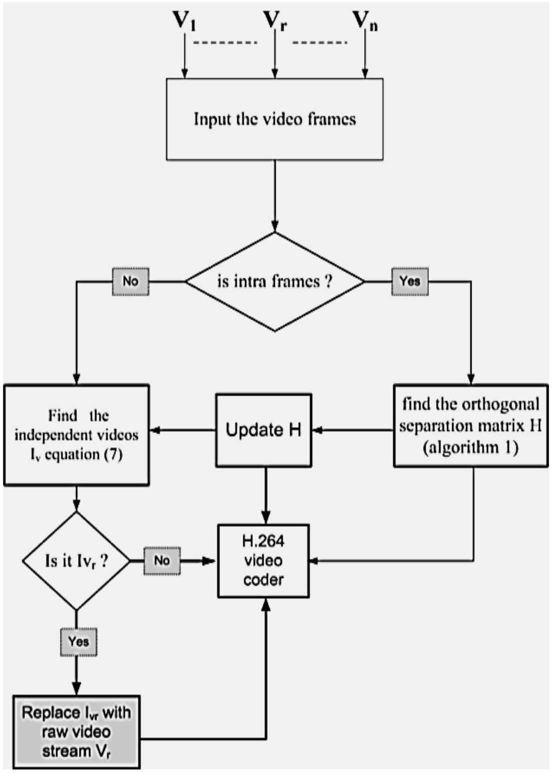

Fig. 3. The proposed I-MVC encoder flowchart.

Finally, the correlations between the video streams are exploited to enhance the quality of the encoded video streams. Since the reference video stream V_r is encoded in pixel domain (Section III.A), V_r will be used as a reference to enhance the other decoded video streams V_i and we call this as the *refinement* algorithm.

The first step in the refinement algorithm is to apply the FastICA algorithm on the intra frames of V_i and V_r . By using FastICA the intra frames of V_i and V_r can be decomposed into a mixing matrix A and independent components S (8). The A matrix (8) is estimated by deploying FastICA on the intra frames of V_i and V_r. Thus, A will contain all the true correlation information between the reference video stream V_r and the other video stream V_i.

$$\begin{bmatrix} V_i \\ V_r \end{bmatrix} = \mathrm{AS} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} S_i \\ S_r \end{bmatrix} \quad (8)$$

where S_i and S_r are the ICs of the video frames V_i and V_r respectively .

The crux of the refinement algorithm is to estimate a new version of each video stream V_i by solving the optimization problem (10). To explain the optimization problem (10) assume the under-determined problem (9) where A and V_r are given and the vector of independent components S is unknown. The problem (10) can be solved depending on the sparse feature of the video frames [16]. However, according to [16] the problem (9) can be formulated into the optimization problem (10). The optimization (10) is used to estimate the independent components $\hat{S}$ with minimum ℓ_1 norm subject to $V_r = [a_{21} \quad a_{22}]\hat{S}$.

$$[V_r] = [a_{21} \quad a_{22}] \begin{bmatrix} S_i \\ S_r \end{bmatrix} \quad (9)$$

$$\hat{S} = arg\,min\, \|\hat{S}\|_1 \;\; S.t \;\; V_r = [a_{21} \quad a_{22}]\hat{S} \quad (10)$$

where $\|.\|_1$ is the ℓ_1 norm

Finally the estimated independent components $\hat{S}$ can be used to find a new version $\hat{V_i}$ (11) of the video frame V_i. Since $\hat{V_i}$ is estimated depending on the maxing matrix A so $\hat{V_i}$ will contain more correlation information about V_r than the old V_i.

$$[\hat{V_i}] = [a_{11} \quad a_{12}] \begin{bmatrix} \hat{S_i} \\ \hat{S_r} \end{bmatrix} \quad (11)$$

The refinement algorithm can be summarized as:

Algorithm 3: Refinement algorithm.

For every two video streams (the reference video stream V_r and any other video stream V_i) do:

3.1 Find the matrix A (Algorithm 1 steps 1.1 to 1.3) from the intra frames of V_r and V_i.

3.2 For all the frames in V_i except the intra frames divide the frames into micro-block (MB) and for every MB_i do:

i. Find the co-located micro-block (co_MB) in the reference video stream V_r.

ii. Find the best matching micro-block (b_MB) in the reference video stream V_r for the MB_i.

3.3 If the b_MB = co_MB End, Else go to 3.4.

3.4 Solve the optimization problem (13) to estimate a new version of the n_MB_i.

3.5 Replace the MB_i in V_i with the average between MB_i and n_MB_i..

The optimization problem (12) is similar to (10) with one additional constraint:

$$\hat{S} = \arg min \|\hat{S}\|_1 \;\; \mathrm{s.t}$$
$$\mathrm{vect}(co\_MBi) = [a_{21} \quad a_{22}]\hat{S} \quad (12)$$
$$\mathrm{vect}(b\_MBi) - [a_{11} \quad a_{12}]\hat{S} \le \varepsilon$$

where vect(.) is the vectorization operator.

The new constraint in (12) is added to guarantee that the difference between the new estimated micro-block (n_MBi) and the best matched MB b_MBi is less than or equal to ε. In turn this ensures that the n_MBi has the correct disparity vector as the MB_i and this will improve the quality of the MB n_MBi.

Finally the optimization problem (12) can be rewritten as:

TABLE I. THE BIT RATE AND THE PSNR OF PROPOSED I-JMVC COMPARE WITH THE CONVENTIONAL JMVC

QP	Data set	JMVC		Proposed I-MVC		Percentage of Saving in bit rate %	Different in PSNR JMVC-(I-MVC)
		Bit rate Kb/s	*PSNR dB*	*Bit rate Kb/s*	*PSNR dB*		
37	*Ballroom*	233.5	32.01	211.2	31.9	9.6	0.11
	Exit	117.5	35	101	34.81	14.1	0.19
	Vassar	117	32.9	107	33.5	8.6	-0.6
	Uli	972.5	32	639.5	31.92	34.2	0.08
	Object	347	32.4	278.3	32.02	19.8	0.38
	Flamenco2	357	32.6	298	32.9	16.5	-0.3
Overall average at QP=32						17.1	-0.024
32	*Ballroom*	428.5	34.7	361.4	34.4	15.7	0.3
	Exit	201.5	37.2	164	36.9	18.6	0.3
	Vassar	248	35.01	192	35.34	22.6	-0.33
	Uli	1724	34.9	1208	34.62	29.9	0.28
	Object	612.3	35.63	442.7	35.5	27.7	0.13
	Flamenco2	649	35.5	533	35.61	17.8	-0.11
Overall average at QP=37						22.1	0.095
27	*Ballroom*	845	37.36	706.9	37.13	16.3	0.23
	Exit	397	39.07	328.1	38.7	17.4	0.37
	Vassar	602	36.9	440	37.2	26.9	-0.3
	Uli	3067	37.58	2423	37.21	21	0.37
	Object	1059	39	797.8	38.8	24.7	0.2
	Flamenco2	1195.5	38.6	1123	38.5	6.1	0.1
Overall average at QP=27						18.7	0.162
22	*Ballroom*	1709.1	39.5	1246.1	39.05	27.1	0.45
	Exit	956.1	40.4	645.1	40.01	32.5	0.39
	Vassar	1572	38.9	967	38.91	38.5	-0.01
	Uli	5601	39.7	4162	39.2	25.7	0.5
	Object	1734.5	41.63	1254.5	41.5	27.7	0.13
	Flamenco2	2147	41.01	1901	40.7	11.5	0.31
Overall average at QP=22						27.2	0.295
The overall average						21.23	0.132

$$\hat{S} = \arg\min \left(\|\hat{S}\|_{\psi 1} + \alpha\, C_1 + \beta\, C_2 \right) \text{ and}$$
$$C_1 = \left\| \mathrm{vect}(co\_MBi) - [a_{21} \quad a_{22}]\hat{S} \right\|_{\psi 2} \qquad (13)$$
$$C_2 = \left\| \mathrm{vect}(b\_MBi) - [a_{11} \quad a_{12}]\hat{S} \right\|_{\psi 2}$$

where the $\|.\|_{\psi i}$ is the ℓi norm in the Ψ domain

Problem (13) can be solved depending on the sparse feature of S [16]. Thus to solve (13) we need to find a convenient transformation (domain) where S is sparse in that domain. In this paper we propose to utilize the (Discrete cosine transform) DCT transformation as the sparse domain.

IV. EXPERIMENTAL RESULTS

In this section, the performance of the proposed I-MVC algorithm will be tested under the common test conditions of [17]. The coding parameters and properties for each attempted video stream data set are given in [17]. We compare the performance of the proposed algorithm with that of the JMVM [4]. For the proposed I-MVC encoder, we assume that the intra frame period of the B picture hierarchical structure is 12 for all tested video sequences. In the test experiment, several standard video sequences are used. For data sets with 8 1D/ parallel views the proposed I-MVC is tested with 3 views. For data sets with 2D/cross arrangement, such as Flamenco2 [17], all five views are used in the test experiment.

The average bit rate and the average peak single to noise ratio (PSNR) for the luminance (Y) component with different quantization parameters (QP) are shown in Table I.

It is observed that our proposed I-MVC algorithm achieves up to 38.5% saving in bit rate with minor loss in the average PSNR compared to the JMVC algorithm. The difference in PSNR between JMVC and I-MVC increases with the increase of the coding bit rate (lower QP). Since in I-MVC the orthogonal separation matrix H is estimated every intra frame period (Algorithm 1), increasing the bit rate can result in missing some of the minor correlation information (details). The influence of this defect appears with lower QP when more details are encoded. Even though, in the worst case the difference in PSNR of the JMVC and the proposed I-MVC does not exceed 0.45 dB (Table I, Ballroom at QP=22). On average, the proposed algorithm I-MVC results in 21%

savings in the overall bit rate with less than 0.15 dB loss in the overall PSNR.

Moreover, Table II shows the reduction in compute time of the proposed I-MVC algorithm compared to that of conventional JMVC. The computational time is measured for one complete GOP during the encoding phase. The experiment on evaluating time complexity is performed with Matlab 2011a package running on a machine with Intel core i7 Q 720@ 1.60 GH/s CPU. Table II shows that the proposed algorithm achieves up to 65% reduction in the computational complexity.

V. CONCLUSION

In this paper, a low complexity 3D multi-view video coder is introduced. The proposed algorithm is based on independent component analysis (ICA) thus called independent multi-view video coding (I-MVC). The proposed I-MVC constitutes of two new algorithms, one for the encoder and one for the decoder. At the encoder, ICA is used to remove correlation and redundancies between the video sequences to produce independent components (ICs). Each ICs is independently encoded by the H.264 video coder. Thus, the complexity of the I-MVC encoder is reduced as I-MVC does not involve the DE processes required by JMVC. Moreover, I-MVC results in a reduction of the transmitted coding bit rates. The I-MVC solves the GOP-PS problem also, as the video streams are independently encoded and don't require a GOP-PS. Our theory is verified by experimental results on publicly available 3D MVV sequences and we show that I-MVC achieves significant reduction in computational complexity and in the transmission bit-rate.

TABLE II. COMPLEXITY ANALYSIS

Data set	Time/SEC		Reduction in the computation time %
	JMVC	*I-MVC*	
Ballroom	580.8	341.1	41.2
Exit	498	252	49.8
Vassar	370.4	197.2	46.8
Uli	1720.2	832.7	51.6
Object	511.2	178.4	65.1
Flamenco2	1319.4	772.9	41.4
Average time saving			49.3

REFERENCES

[1] A. Kubota, A. Smolic, M. Magnor, et. al, "Multiview imaging and 3dtv," *Signal Processing Magazine, IEEE*, vol. 24, no. 6, pp. 10 –21, nov. 2007.

[2] A. Smolic, K. Mueller, P. Merkle, et. al, "Free viewpoint video extraction, representation, coding, and rendering," in *Image Processing, ICIP'04. 2004* vol. 5. IEEE, pp. 3287–3290.

[3] A. Smolic, K. Mueller, N. Stefanoski, et. al,, "Coding algorithms for 3dtv—a survey," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 11, pp. 1606 –1621, nov. 2007.

[4] A. Vetro, P. Pandit, H. Kimata, et. al, "Joint multiview video model (JMVM) 8.0," Doc. JVT-AA207, Geneva, Switzerland, April 2008.

[5] P. Merkle, A. Smolic, K. Muller, et. al, "Efficient prediction structures for multiview video coding," *Circuits and Systems for Video Technology, IEEE Trans. on*, vol. 17, no. 11, pp. 1461–1473, 2007.

[6] Y. Chen, S. Liu, Y. Wang, et. al,"Low-complexity asymmetric multiview video coding," in *Multimedia and Expo, 2008 IEEE International Conference on*. 2008, pp. 773–776.

[7] L. Shen, Z. Liu, T. Yan, et. al, "View-adaptive motion estimation and disparity estimation for low complexity multiview video coding," *Circuits and Systems for Video Technology, IEEE Trans.* vol. 20, no. 6, pp. 925–930, 2010.

[8] H. S. Hussein, M. El-Khamy, and M. El-Sharkawy, "Blind configuration of multi-view video coding," in *Consumer Electronics (ICCE), 2012 IEEE International Conference on*, 2012, pp. 392 –393.

[9] Y. Kim, J. Kim, and K. Sohn, "Fast disparity and motion estimation for multi-view video coding," *Consumer Electronics, IEEE Trans.* vol. 53, no. 2, pp. 712–719, 2007.

[10] X. Li, D. Zhao, X. Ji, et. al, "A fast inter frame prediction algorithm for multi-view video coding," in *Image Processing, ICIP IEEE International Conference on*, vol. 3. 2007, pp. III–417.

[11] Y. Zhang, G. Jiang, M. Yu, and Y. Ho, "Adaptive multiview video coding scheme based on spatiotemporal correlation analyses," *ETRI journal*, vol. 31, no. 2, pp. 151–161, 2009.

[12] G. J. et al., "Multi-modal multi-view video coding based on correlation analysis," *Chinese Journal of Computer*, vol. 30(12), pp. pp. 2205–2211, 2007.

[13] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent component analysis*. Wiley-interscience, 2001,

[14] A. Hyvarinen and E. Oja, "A fast fixed-point for independent component analysis," *Neural computation*, vol. 9, no. 7, pp. 1483–1492, 1997.

[15] J. Cardoso, "Multidimensional independent component analysis," in *Acoustics, Speech and Signal Processing, IEEE International Conference on*, vol. 4. 1998, pp. 1941–1944.

[16] E. Candes and J. Romberg, "l1-magic: Recovery of sparse signals via convex programming," *URL: www. acm. caltech. edu/l1magic/downloads/l1magic.*

[17] A. Vetro, Y. Su, and A. Smolic, "Common test conditions for multiview video coding,", JVT-T207, Klagenfurt, Austria, Jul, 2006.