A Novel Rate-Distortion Method in 3D Video Capturing in the Context of High Efficiency Video Coding (HEVC) in Intelligent Communications

Ioannis M. Stephanakis[1], Ioannis P. Chochliouros[2], Anastasios Dagiuklas[3], and George C. Anastassopoulos[4]

[1] Hellenic Telecommunication Organization S.A. (OTE), 99 Kifissias Avenue, GR-151 24, Athens, Greece
stephan@ote.gr
[2] Research Programs Section, Hellenic Telecommunication Organization S.A. (OTE) 99 Kifissias Avenue, GR-151 24, Athens, Greece
ichochliouros@oteresearch.gr
[3] Hellenic Open University, Parodos Aristotelous 18, GR-262 22, Patras, Greece
ntan@teimes.gr
[4] Democritus University of Thrace, Medical Informatics Laboratory, GR-681 00, Alexandroupolis, Greece
anasta@med.duth.gr

Abstract. *High-Efficiency Video Coding* is currently proposed as the newest and most efficient video coding standard by the ITU-T *Video Coding Experts Group* and the ISO/IEC *Moving Picture Experts Group*. Compression improvement relative to existing standards is estimated in the range of 50%. It is a block-based hybrid video coding algorithm that introduces several novel features compared to MPEG-4 like *Coding Units* associated with *Prediction Units* and *Transform Units*, *Advanced Motion Vector Prediction*, minimization of a rate-distortion Lagrangian cost, directional orientations for intra-picture prediction etc. The core algorithm of *Context-Adaptive Binary Arithmetic Coding* is based upon that of the MPEG-4 standard. View synthesis algorithms for HEVC stereo and 3D encoding are expected to be finalized as a standard extension. A *Multi-view Video Coding* scheme based upon the estimation of correlated parameter sets of elastic models between views is wherein adopted. The order of the tensor equals the number of multiple views. Underlying distributions are updated step-by-step. They are modeled according to context indices.

Keywords: object oriented coding, H264/AVC, High Efficiency Video Coding HEVC/H.265, higher order motion compensation models, CABAC, 3D television, *Rate-Distortion* theory.

1 Introduction

High Efficiency Video Coding (HEVC) [1,2] is a video compression standard that is proposed as a successor to H.264/MPEG-4 AVC (*Advanced Video Coding*) developed

H. Papadopoulos et al. (Eds.): AIAI 2013, IFIP AICT 412, pp. 462–473, 2013.

by the ISO/IEC Moving Picture Experts Group (MPEG) and the ITU-T Video Coding Experts Group (VCEG) under the name ISO/IEC 23008-2 MPEG-H Part 2 and ITU-T H.HEVC [3]. MPEG and VCEG have established a Joint Collaborative Team on Video Coding (JCT-VC) in order to develop the HEVC standard. HEVC standard is said to improve video quality, double the data compression ratio compared to H.264, and support 8K UHD and resolutions up to 8192×4320 pixels [4]. Two profiles are supported, namely Main Profile and High Efficiency 10 (HE10). The HEVC Main Profile (MP) is compared in coding efficiency to H.264/MPEG-4 AVC High Profile (HP), MPEG-4 Advanced Simple Profile (ASP), H.263 High Latency Profile (HLP), and H.262/MPEG-2 Main Profile (MP). As of the end of January 2013, ISO/IEC and ITU-T had approved HEVC as ISO/IEC 23008-2 High Efficiency Video Coding and ITU-T Rec. H.265 respectively, as the final draft international standard.

Simulcast as well as Multiview Video Coding are described in the context of the MPEG-4 AVC standard (Version 8: July 2007 (including SVC extension), Version 9: July 2009 (including MVC extension). [5]. A standard MVC coder consists of *N* parallelized single view coders. Each of them uses temporal prediction structures and encodes a sequence of successive pictures as intra (I), predictive (P) or bi-predictive (B) frames. Nevertheless MVC is not appropriate for delivering 3-D content for autostereoscopic displays since the bit rate required for coding multiview video with the MVC extension of H.264/AVC increases approximately linearly with the number of coded views. The transmission of 3D video in the Multiview Video plus Depth (MVD) format appears as a promising alternative. ISO/IEC and ITU-T established JCT-3V for the 3D video coding standards as of July 2012. AVC compatible video-plus-depth extension is expected within 2013 [6]. The transmission of 3D video in according to Multiview Video plus Depth (MVD) associates encoded depth data representing the basic geometry of the captured video scene with picture frames. Depth data are estimated based on the acquired pictures. They are obtained by the application of depth estimation algorithms and should not be regarded as ground truth. Based on the transmitted video pictures and depth maps, additional views suitable for displaying 3D video content on autostereoscopic displays can be generated using depth image based rendering (DIBR) techniques at the receiver side. All video pictures and depth maps that represent the video scene at the same time instant build an access unit. The access units of the input MVD signal are coded consecutively similar to MVC. The video picture of the so-called independent view inside an access unit is transmitted first directly followed by the associated depth map. Thereafter, the video pictures and depth maps of other views are transmitted. A video picture is always directly followed by the associated depth map. *Advanced Motion Vector Prediction* (AMVP) is used. Work on HEVC 3D and scalable extensions has been currently under development. It focuses on

- Coding of Independent Views (2D video coding)
- Coding of Dependent Views (DCP, inter-view motion/residual prediction)
- Coding of Depth Maps
- Encoder Control
- View synthesis algorithms

Final draft amendment (FDAM) for HEVC for stereo and 3D is expected in 2015. It improves the compression capabilities for dependent video views and depth data. As of the end of January 2013, ISO/IEC and ITU-T had approved HEVC as ISO/IEC 23008-2 High Efficiency Video Coding and ITU-T Rec. H.265 respectively, as final draft international standard.

2 Higher Order Models for Motion Compensation and Encoder Control

2.1 Novel Video Encoding Features of the HEVC/H. 265 Standard

The video coding layer of HEVC employs the same "hybrid" approach used in all video compression standards since H.261. Each picture is split into block-shaped regions, with the exact block partitioning being conveyed to the decoder. The first picture of a video sequence (and the first picture at each "clean" random access point into a video sequence) is coded using only intra-picture prediction. Transform coefficients from region-to-region within the same picture are spatially predicted but there are no dependencies upon other pictures. The remaining pictures of a sequence or frames between random access points are encoded using temporally-predictive coding modes. The encoding process for inter-picture prediction consists of choosing motion data comprising the selected reference picture and motion vectors (MV) that are applied for predicting the samples of each block. Residual frames are transformed by a linear spatial transform. The transform coefficients are scaled, quantized, entropy coded and transmitted together with the prediction information. Novel features of the HEVC standard are outlined as follows:

- Larger and more flexible coding, prediction, and transform units.
- Improved mechanisms to support parallel encoding & decoding.
- More flexible temporal prediction and scanning structure.
- More accurate intra prediction approach and directions/modes.
- More accurate motion parameters (including merge mode) and sub-pixel prediction.
- Inclusion of non-square transform and allowing asymmetric motion prediction.
- More flexible transform, choice of DST, and no-transform option.
- Rate-distortion optimized quantization (RDOQ).
- Improved in-loop filters, including the new sample adaptive offset

The *Coding Tree Unit* (CTU) in HEVC replaces the *macroblock* structure as known from previous video coding standards. It has a size selectable by the encoder and it can be larger than a traditional *macroblock*. The quadtree syntax of the CTU (see Fig. 1) specifies the size and positions of its luma and chroma *Coding Blocks* (CB). A *Coding Tree Block* (CTB) may contain only one *Coding Unit* (CU) or may be split to form multiple CUs. Thus each CU is characterized by its *Largest* CU (LCU) size and the hierarchical depth in the LCU that the CU belongs to. It has an associated partitioning into *Prediction Units* (PUs) and a tree of *Transform Units* (TUs).

Intra-picture prediction, inter-picture prediction or skip mode are selected at CU level. A PU is basically the elementary unit for prediction and it is defined after the last level of CU splitting. Prediction type and PU splitting type are two concepts that describe the prediction method. Intra prediction allows for symmetric splitting whereas inter prediction allows for both symmetric and asymmetric splitting. At the level of PU, intra-prediction is performed from samples already decoded in adjacent PUs. Such modes as DC (average/flat), angular directions (one from up to 33 as in Fig. 2), *Planar Intra Prediction* (surface fitting), SDIP (*Short Distance Intra Prediction*), MDIS (*Mode Dependent Intra Smoothing*) may be used. Advanced motion prediction (see for example [7]) featuring a "merge" mode or the skip mode may be used for inter prediction. Quarter-pixel precision and 7-tap or 8-tap filters are used for interpolation of fractional-sample positions. Multiple reference pictures are used. A deblocking filter similar to the one used in H.264/MPEG-4 AVC is operated in the inter-picture prediction loop. An adaptive loop filter (ALF) is alternatively employed for higher efficiency. The coefficients of ALF are calculated and transmitted on a frame basis and the MMSE estimator is used. For each degraded frame, ALF can be applied to the entire frame or to local areas. Similar transforms as for H.264 (including the discrete sine transform and the Hadamard transform) are used for encoding the residual data. Three different scanning modes (namely zigzag, horizontal and vertical scan) are used to improve the residual coding.

Such novel features as the option to partition a picture into rectangular regions called *tiles* and *wavefront parallel processing* (WPP) are introduced into the HEVC standard in order to enhance its parallel processing capabilities.

The core algorithm of *Context-Adaptive Binary Arithmetic Coding* (CABAC) is based upon that of the H.264/MPEG-4 standard. The number of contexts used in HEVC is substantially less than the number of contexts in H.264/MPEG-4 AVC whereas the entropy coding design allows for better compression.

2.2 Motion Compensation Using Higher Order Models and Rate-Distortion Control

Higher order motion models and view synthesis techniques are currently investigated in the literature for sub-pixel motion compensation that may be applied 3D and multiview encoding. Higher order motion models such as the *affine model* (AMMCP) [8], the *mesh based* MCP [9], the *elastic* MCP [10] and *View Synthesis by Depth Image Rendering* (DIBR) [11, 12] have been proposed as possible extensions of existing standards. The transformation of one view to another is based on the camera parameters. DIBR uses a view and a depth map for generating arbitrary views. Rough 3D information can be reconstructed based on the depth map so that the transformation due to the disparity effect can be generated. The video compression with DIBR becomes the compression of depth image instead of the correlations between views. The elastic MCP model estimates the spatial transformation parameters between the predicted block with translational MV and the current block. The motion parameters are encoded in the bit stream. The pixel location of the predicted block (x_i, y_i) is transformed to (x'_i, y'_i) by the following equations,

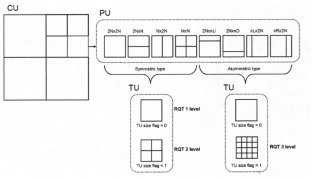

Fig. 1. Structure of Coding Tree Units (CTU). Prediction and Transform Units (PU and TU) are depicted.

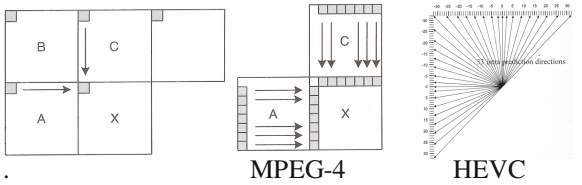

Fig. 2. Intra-picture prediction for boundary samples of adjacent blocks (MPEG-4 and HEVC)

$$x'_i = x_i + \sum_{l=1}^{P/2} m_l \varphi_l(x_i, y_i) \quad \text{and} \quad y'_i = y_i + \sum_{l=P/2+1}^{P} m_l \varphi_l(x_i, y_i) \tag{1}$$

where P is the number of elastic parameters used, m_l is the elastic parameter and $\varphi_l(x_i, y_i)$ is the basis function. The elastic MCP model uses discrete cosines as the basis functions and it is defined by

$$\varphi_l(x_i, y_i) = \varphi_{l+P/2}(x_i, y_i) = \cos\left(\frac{(2x_i+1)\pi u}{2\,blocksize\ x}\right)\cos\left(\frac{(2y_i+1)\pi v}{2\,blocksize\ y}\right), \tag{2}$$

where $l = \sqrt{P/2}\ u + v + 1$ and $0 \le (u, v) \le \sqrt{P/2} - 1$. Conventional motion vector is equal to motion parameters [m_1 $m_{P/2+1}$] in such a case.

The minimization of a Lagrangian cost function for motion estimation was proposed in [13] (see Fig. 3). Given a reference picture list R and a candidate set M of motion vectors, the motion parameters for a block $\boldsymbol{s}_k$, which consists of a displacement or motion vector m=[m_x, m_y] and, if applicable, a reference index r, determine the coding mode for coding a block of samples, such as a *macroblock* or a *coding unit*. Additional features of a coding mode are the intra or inter prediction modes or partitions for motion-compensated prediction or transform coding including the quantization step. Given the set of applicable coding modes for a block of samples $\mathbf{s}_k$, the used coding mode is chosen according to

$$c^* = \arg\min_{c \in C_k} (D_k(c) + \lambda \cdot R_k(c)) \tag{3}$$

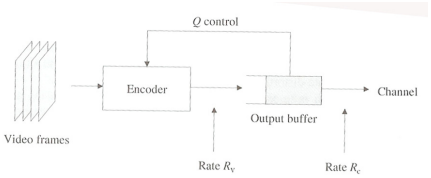

Fig. 3. Rate control model

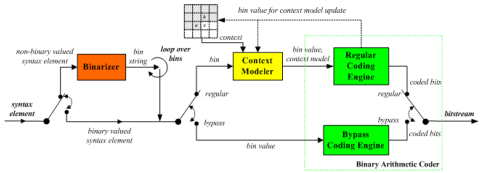

Fig. 4. Context-based Adaptive Binary Arithmetic Coding

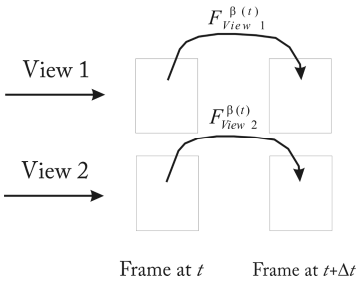

Fig. 5. Updates-update symbols between sequential video frames

where the distortion term $D_k(c)$ represents the SSD between the original block $\mathbf{s}_k$ and its reconstruction $\mathbf{s}'_k$, that is obtained by coding the block $\mathbf{s}_k$ with the mode c. The term $R_k(c)$ represents the number of bits (or an estimated thereof) that are required for representing the block $\boldsymbol{s}_k$ using the coding c for the given bitstream syntax. It includes the bits required for signaling the coding mode and the associated side information (e.g. motion vectors, reference indices, intra prediction modes and coding modes for sub-blocks of $\boldsymbol{s}_k$) as well as the bits required for transmitting the transform coefficient levels representing the residual signal. A coding mode is often associated with additional parameters such as coding modes for sub-blocks, motion parameters and transform coefficient levels. While coding modes for sub-blocks are determined in advance, motion parameters and transform coefficient levels are chosen according to Eq. 3. For calculating the distortion and rate terms for the different coding modes, decisions for already coded blocks of samples are taken into account (e.g. by considering the correct predictors or context models).

3 A Scalable Context-Based Adaptive Model for Encoding Motion Compensation Parameters in Multi-view 3D Systems

3.1 The Context-Based Adaptive Binary Arithmetic Coding (CABAC) Model

CABAC (*Context-based Adaptive Binary Arithmetic Coding*) as well as *Rate-Distortion Optimization* are included in the HEVC standard. *Context-based Adaptive Binary Arithmetic Coding* (CABAC) [14] achieves good compression performance through the selection of probability models for syntax elements according to context. It estimates adaptively probabilities based on local statistics and uses arithmetic coding rather than variable-length coding. The following steps [15,16,17,18] are involved in coding a data symbol: 1. *Binarisation*, 2. *Context model selection*, 3. *Arithmetic encoding* and 4. *Probability update.* The above steps are

illustrated in Fig. 4. The design of binarization schemes relies on a few basic code trees, whose structure enables a simple on-line computation of all code-words without the need for storing tables. There are four such basic types [16] namely the *unary code*, the *truncated unary code*, the *k*-th *order Exp-Golomb code* and the *fixed-length code*. There are binirization schemes based on a concatenation of these elementary types. The nodes located in the vicinity of the root node of a binary tree are the natural candidates for being modelled individually, whrereas a joint model should be assigned to all nodes on deeper tree levels corrresponding to the tail of the probability density function. According to CABAC, given a predefined set $\boldsymbol{T}$ of past symbols, a so-called *context template*, and a related set $\boldsymbol{C}$={0,…,*C*-1} of contexts, contexts are specified by a modeling function $F:T\to C$ operating on the template $\boldsymbol{T}$. For each symbol x to be coded, a conditional probability $p(x|F(z))$ is estimated by switching between different probability models according to the already coded neighboring symbols $z\in T$. After encoding x using the estimated conditional probability $p(x|F(z))$, the probability model is updated with the value of the encoded symbol x. Thus $p(x|F(z))$ is estimated on the fly by tracking the actual source statistics. The entity of probability models used in CABAC can be arranged in a linear fashion such that each model can be identified by a unique so-called context index γ.

3.2 A Context-Based Adaptive Model for Correlated Motion Compensation Parameters in Multiple Views

The proposed encoding scheme investigates context-based adaptive models for correlated multilinear object entities between multiple views [19]. It is a model that does not depend upon the parameters of the recording cameras. It is applicable to such object entities as transform elements and/or motion vector parameters and it scales in a straightforward fashion from stereo to multiple free views. It combines adaptivity regarding the context templates with updates of the correlated orthonormal features of the object entities through all views per GOP. Let us assume an N^{th}-order tensor A, which resides in the tensor multi-linear space $R^{I_1}\otimes R^{I_2}\otimes\cdots\otimes R^{I_N}$ where $R^{I_1},R^{I_2}\cdots,R^{I_N}$ are the N vector linear spaces of multiview system featuring N views. The "k-mode vectors" of A are defined as the I_k-dimensional vectors obtained from A by varying its index in k-mode while keeping all other indices fixed [20,21]. Multi-view motion prediction using cross-view prediction vectors may be defined per GOP through a video object that we call *Motion Prediction video-Object* and is denoted as **MPO**. Let us define a group of motion vector parameters pertaining to k-view within some GOP as $\mathbf{M}^{(k)}=\left[M_1^k \quad M_2^k \quad \cdots \quad M_{I(k)}^k\right]$. Prediction of motion vectors - which are denoted as $\mathbf{v}(m_{block},n_{block},t)=[v_x\ v_y]$ for a block indexed by (m_{block},n_{block}) at t - is carried out once with respect to one of the views or a linear combination of a selected subset. The motion vectors pertaining to k-view according to the elastic model may be decomposed as follows,

$$\mathbf{v}^k(m_{block},n_{block},t)=\mathbf{v}(m_{block},n_{block},t)+[\sum_{l=1}^{P/2}m_l^k\varphi_l \quad \sum_{l=P/2+1}^{P}m_l^k\varphi_l]=[v_x\ v_y]+\sum_{i=1}^{I(k)}\alpha_i^k(m_{block},n_{block},t)M_i^k\cdot \quad (4)$$

Motion Prediction video-Object (**MPO**) holds the orthonormal cross-view prediction vectors. It is defined as,

$$MPO(m_{block}, n_{block}, t) = \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_N=1}^{I_N} s(i_1, i_2, ...; t)\, M_{i_1}^1 \circ M_{i_2}^2 \cdots M_{i_N}^N = S(t) \times_1 \mathbf{M}^{(1)} \times_2 \mathbf{M}^{(2)} \times_3 \mathbf{M}^{(3)} \times \cdots \times_N \mathbf{M}^{(N)} \quad (5)$$

Unfolding *MPO* along the *k*-mode is denoted as

$$\mathbf{MPO}_{(k)} \in R^{I_k \times (I_1 \times \cdots \times I_{k-1} \times I_{k+1} \times \cdots \times I_N)}, \quad (6)$$

where the column vectors of $MPO_{(k)}$ are the *k*-mode vectors of *MPO*. Unfolding the *Multi-view Video Plane* of an *N*-view system along the *k*-mode view results into the following matrix representation

$$\mathbf{MPO}_{(k)}(t) = \mathbf{M}^{(k)} . S_{(k)}(t) . \left(\mathbf{M}^{(k+1)} \otimes \mathbf{M}^{(k+2)} \otimes \cdots \otimes \mathbf{M}^{(N)} \otimes \mathbf{M}^{(1)} \otimes \cdots \otimes \mathbf{M}^{(k-1)}\right)^{\mathsf{T}}, \quad (7)$$

where $\otimes$ denotes the Kronecker product. The core tensor *S* (in a representation similar to the one described in Eq. 8) is analogous to the diagonal singular value matrix of the traditional SVD and coordinates the interaction of matrices to produce the original tensor. Matrices $\mathbf{M}^{(k)}$ are orthonormal and their columns span the space of the corresponding flattened tensor denoted as $\mathbf{MPO}_{(k)}$. The objective of MPC analysis for predetermined dimensionality reduction is the estimation the *N* projection matrices $\{\tilde{\mathbf{M}}^{(k)}(t) \in R^{I_k \times P_k}, k = 1, ..., N\}$ that maximize the total tensor scatter [22],

$$\left\{\tilde{\mathbf{M}}^{(k)}(t), k = 1, ..., N\right\} = \arg \max_{\tilde{\mathbf{M}}^{(1)}, \tilde{\mathbf{M}}^{(2)} ... \tilde{\mathbf{M}}^{(N)}} \left\| MPO(t) - \Delta(t) \right\|^2, \quad (8)$$

where $\sum_{k=1}^{N} P_k \le c$. One may estimate adaptively the cross products between motion vector parameters of different video views by maximizing sequentially Eq. 8 for all modes. Its decomposition into non-negative parts and the application of a multiplicative update rule that maintains orthonormality [23,24] is suggested in [19].

The following algorithm outlines in detail the steps of the proposed encoding approach for multi-view systems:

Select reference frames for GOP and initialize motion vector parameters per GOP

I – Estimate the average motion vectors [v_x v_y] that are common for all views according to Eq. 4.

II – Subtract average values and estimate the sets of motion vector parameters that are strongly correlated. Carry out prediction using adjacent macroblocks and solve Eq. 8 iteratively (on the fly) for optimal **MPOs** using multiplicative updates (or operational updates). Assume (Fig. 5) that

$$\tilde{\mathbf{M}}^{(k)}(t + \Delta t) = \mathbf{\Phi}_{View\ k}(t + \Delta t, t)\tilde{\mathbf{M}}^{(k)}(t) + \mathbf{b}(t)\mathbf{w}(t) \cong F_{View\ k}^{\beta(t)}(\tilde{\mathbf{M}}^{(k)}; t + \Delta t, t) + \mathbf{b}'(t)\mathbf{w}(t) \quad (9)$$

where *t* is the frame index, $\boldsymbol{w}(t)$ stands for a white process featuring zero mean and (t)denotes

some type of operation at t like *element-by-element multiplication*, *component rotation*, *permutation* etc.

III- Initial sets of motion vector parameters are transmitted as ordered mutivalue sequences indexed by position/order of component. Probability density functions are entropy encoded. A *context index increment* denoted as $\chi_s(order\_of\_component, bin\_index)$ can be assigned. Select the context template for the $\sum_{k=1}^{N} P_k \leq c$ most correlated linear projections of motion parameter sets between views, i.e the joint pdfs yielding the highest values of $S(i_1, i_2, \ldots; t)$. Address *order_of_component* starting from the maximum S in decreasing order. This constitutes an indirect indexing of orthonormal basis vectors. Transmit symbols $\bullet(t)$ (multiplicative updates or operations) per view and *order_of_component* in a similar fashion (up to P_k indices for View k) as indicated by the structure of **MPOs**.

IV- Encode the symbol levels corresponding to the values of the selected sets of the motion vector parameters, i.e. for the set yielding the smallest distortion error provide *order_of_component* and level. Each level is encoded as a sequence of indexed bins using CABAC models. Find residual distortions.

V- Carry out rate control according to Eq. 3 by selecting the total number of symbols. Select between the c most correlated symbols per block/CTU. Continue as long as one gets a valid estimate, i.e. maintain decreasing distortion differences. Otherwise break.

VI- Encode residual frames and **repeat until end of GOP** (go to *Step* II).

4 Numerical Simulations

Numerical simulations for the proposed encoding method have been carried out for the image sequences used for video view interpolation as described in [25]. Each sequence is 100 frames long. The camera resolution is 1024x768 and the capture rate is 15fps. The frames of the ballet sequence in [25] are used to obtain the numerical values presented in this section. The multi-view GOP for the numerical simulations

consists of initial four (4) frames. A fixed size macroblock featuring dimensions of 8x8 pixels is used to carry out motion compensation. The average of the first frame of View 1 and of the first frame of View 2 is used as a reference. The elastic MCP model as described in Eq. 2 uses sixteen (16) discrete cosines as the basis functions (parameter *P* equals 32). The context and the distributions for the first stereo GOP of the sequence are illustrated in Fig. 6 Entropy encoding is used based upon the probabilities depicted in Fig. 6.a. The 32x32 sigma values ordered according to magnitude as obtained from the proposed algorithm are given in Fig. 6.b. The histogram of the highest sigma values over all macroblocks is presented in Fig. 6.c. The magnitude of the residual frames corresponding to *View* 1 and *View* 2 are marginally decreased as linear correlated components between views are taken into account. *Peak-Signal-to-Noise-Ratio* (PSNR) values corresponding to the two views are 31.9 dB and 31.0 dB respectively when the two (c=2) most correlated linear components are used. The additional overhead is estimated to 0.11 bits/pixel. Original and residual frames are given in Fig. 7. PSNR values are slightly improved when the six (c=6) most correlated linear components are taken into account. PSNR value for *View* 1 equals 31.9 dB whereas PSNR value for *View* 2 equals 31.2 dB for an estimated additional overhead of 0.17 bits/pixel. Nevertheless savings on the bits/pixel required for underlying motion compensation could be made should the proposed approach be applied.

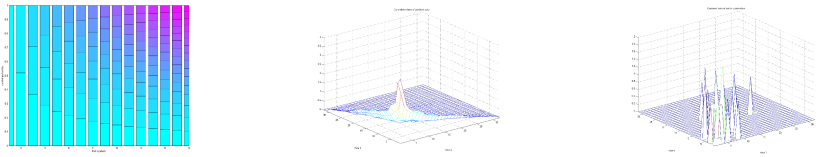

Fig. 6. Contexts and distributions for the $s(i_1, i_2,\ldots;t)$ values for the first four frames of the ballet video sequence (-a- probabilities for one up to sixteen correlation coefficients; -b- the distribution of all 32x32 sigma correlation coefficients and -c- histogram of the highest sigma values over all macroblocks)

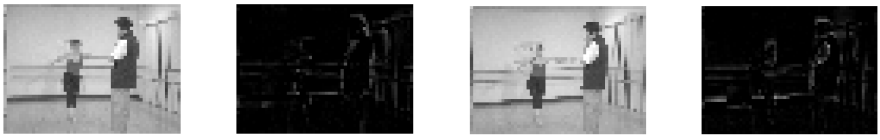

Fig. 7. Views 1 and 2 and residual frames for joint motion vector compensation (the two largest $s(i_1, i_2,\ldots;t)$ values have been used)

5 Conclusion

A novel scalable approach to multi-view video encoding based on the so-called MPO structure is proposed. It takes advantage of the correlated linear components between views and it requires no prior knowledge of the capturing cameras in space. Strongly correlated linear components, i.e. lower order projections of tensorial objects, are

multiplicative per GOP (or frame or slice). The correlated linear components are assigned initial values per GOP (or frame or slice) whereas subsequent operations (updates and permutations) are defined upon index position. Context templates corresponding to different distributions of the sigma correlation parameters may be selected according to the CABAC model. Updating adaptively indexed orthonormal basis functions in conjunction with CABAC increases encoding efficiency and allows for the incorporation of SVD and MPC transforms into the existing approaches. Initial numerical results indicate that the proposed method yields improvements in encoding multiple views in MPEG standards under development like the HEVC.

Acknowledgments. This research work has been funded by the *LiveCity* European Research Project supported by the Commission of the EC – *Information Society and Media Directorate General* (FP7-ICT-PSP, GA No.297291).

References

1. Ohm, J.-R., Sullivan, G.J., Schwarz, H., Tan, T.-K., Wiegand, T.: Comparison of the Coding Efficiency of Video Coding Standards – Including High Efficiency Video Coding (HEVC). IEEE Trans. on Circuits and Systems for Video Technology 22(12), 1669–1684 (2012)
2. Sullivan, G.J., Ohm, J.-R., Han, W.-J., Wiegand, T.: Overview of the High Efficiency Video Coding (HEVC) Standard. IEEE Trans. on Circuits and Systems for Video Technology 22(12), 1649–1668 (2012)
3. ITU TSB (2010-05-21): Joint Collaborative Team on Video Coding. ITU-T, `http://www.itu.int/ITU-T/studygroups/com16/jct-vc/` (retrieved August 24, 2012)
4. `http://www.h265.net/`
5. 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), Version 1: May 2003, Version 2: May 2004, Version 3: March 2005 (including FRExt extension), Version 4: September 2005, Version 5 and Version 6: June 2006, Version 7: April 2007, Version 8: July 2007 (including SVC extension), Version 9: July 2009 (including MVC extension)
6. Vetro, A., Wiegand, T., Sullivan, G.J.: Overview of the Stereo and Multiview Video Coding Extensions of the H.264/MPEG-4 AVC Standard. Proceedings of the IEEE (Special Issue on 3D Media and Displays) 99(4), 626–642 (2011)
7. Kordasiewicz, R.C., Gallant, M.D., Shirani, S.: Affine Motion Prediction Based on Translational Motion Vectors. IEEE Trans. Circuits Syst. Video Technology 17(10), 1388–1394 (2007)
8. Wiegand, T., Steinbach, E., Girod, B.: Affine Multipicture Motion-Compensated Prediction. IEEE Trans. Circuits Syst. Video Technology 15(2), 197–209 (2005)
9. Nakaya, Y., Harashima, H.: Motion Compensation Based on Spatial Transformations. IEEE Trans. Circuits Syst. Video Technol. 4(3), 339–356, 366–367 (1994)
10. Pickering, M.R., Frater, M.R., Arnold, J.F.: Enhanced Motion Compensation Using Elastic Image Registration. In: Proceedings of IEEE Int. Conference on Image Processing, Atlanta, GA, USA, pp. 1061–1064 (2006)
11. Fehn, C.: A 3D-TV Approach Using Depth-Image-Based Rendering (DIBR). In: Proceedings of Visualization, Imaging And Image Processing (VIIP), pp. 482–487 (2003)

12. Schwarz, H., Bartnik, C., Bosse, S., Brust, H., Hinz, T., Lakshman, H., Marpe, D., Merkle, P., Müller, K., Rhee, H., Tech, G., Winken, M., Wiegand, T.: 3D Video Coding Using Advanced Prediction, Depth Modeling, and Encoder Control Methods. In: IEEE Intl. Conf. on Image Processing (October 2012)
13. Sullivan, G.J., Baker, R.L.: Rate-Distortion Optimized Motion Compensation for Video Compression Using Fixed or Variable Size Blocks. In: Proc. of GLOBECOM 1991, pp. 85–90 (1991)
14. Marpe, D., Blattermann, G., Wiegand, T.: Adaptive Codes for H.26L, ITU-T SG 16/6 Document VCEG-L13, Eibsee, Germany (January 2001)
15. Schwarz, H., Marpe, D., Wiegand, T.: CABAC and slices, JVT document JVT-D020, Klagenfurt, Austria (July 2002)
16. Marpe, D., Schwarz, H., Wiegand, T.: Context-Based Adaptive Binary Arithmetic Coding in the H.264 / AVC Video Compression Standard. IEEE Transactions on Circuits and Systems for Video Technology 13(7), 620–636 (2003)
17. Richardson, I.E.G.: H.264 and MPEG-4 Video Compression – Video Coding for Next Generation Multimedia, pp. 198–207. John Wiley and Sons (2003)
18. Marpe, D., Schwarz, H., Wiegand, T.: Probability Interval Partitioning Entropy Codes. Submitted to IEEE Transactions on Information Theory (June 2010), `http://iphome.hhi.de/marpe/download/pipe-subm-ieee10.pdf`
19. Stephanakis, I.M., Anastassopoulos, G.C.: A Multiplicative Multi-linear Model for Inter-Camera Predictio. In: Free View 3D Systems. Engineering Intelligent Systems Journal (under print, 2013)
20. de Lathauwer, L., de Moor, B., Vandewalle, J.: A Multilinear Singular Value Decomposition. SIAM Jour. of Matrix Analysis and Appl. 21(4), 1253–1278 (2000)
21. Lu, H.K.N., Plataniotis, K.N., Venetsanopoulos, A.N.: MPCA: Multilinear Principal Component Analysis of Tensor Objects. IEEE Trans. on Neural Networks 19(1), 18–39 (2008)
22. Lu, H.K.N., Plataniotis, K.N., Venetsanopoulos, A.N.: A Survey of Multilinear Subspace Learning for Tensor Data. Pattern Recognition 44(7), 1540–1551 (2011)
23. Yang, Z., Laaksonen, J.: Multiplicative Updates for Non-negative Projections. Neurocomputing 71(1-3), 363–373 (2007)
24. Zhang, Z., Jiang, M., Ye, N.: Effective Multiplicative Updates for Non-negative Discriminative Learning in Multimodal Dimensionality Reduction. Artificial Intelligence Review 34(3), 235–260 (2010)
25. Zitnick, C.L., Kang, S.B., Uyttendaele, M., Winder, S., Szeliski, R.: High-quality video view interpolation using a layered representation. In: ACM SIGGRAPH and ACM Trans. on Graphics, Los Angeles, CA, pp. 600–608 (2004)