Maximum likelihood motion compensation for distributed video coding

Frederik Verbist\textsuperscript{a,b}, Nikos Deligiannis\textsuperscript{a,b}, Marc Jacobs\textsuperscript{a,b}, Joeri Barbarien\textsuperscript{a,b}, Peter Schelkens\textsuperscript{a,b} and Adrian Munteanu\textsuperscript{a,b,*}  
\textsuperscript{a}Department of Electronics and Informatics, Vrije Universiteit Brussel, Brussels, Belgium  
\textsuperscript{b}Department of Future Media and Imaging, Interdisciplinary Institute for Broadband Technology, Ghent, Belgium

Abstract. Aspiring to provide robust low-complexity encoding for video, this work presents a hash-based transform domain distributed video codec, featuring a novel maximum likelihood motion compensation technique to generate high quality side information at the decoder. A simple hash is employed to perform overlapped block motion estimation at the decoder, which produces a set of temporal predictors on a pixel basis. For every pixel position, maximum likelihood motion compensation, based on an online estimate of the conditional dependencies between the temporal predictors and the original frame, combines the cluster of temporal predictors into a single value to serve as decoder-generated side information. Efficient low-density parity-check accumulate channel codes refine the side information in the transform domain. Experimental results demonstrate that the proposed system advances over our previous hash-based distributed video coding architectures, delivering state-of-the-art distributed coding performance, in particular for sequences organized in large groups of pictures or containing highly irregular motion. Notwithstanding the presence of a hash, the presented distributed video codec successfully maintains low encoder complexity.

Keywords: Distributed video coding, Wyner-Ziv video coding, side information generation, hash-based motion estimation, motion compensation

1. Introduction

The information theoretic principles of distributed source coding (DSC) coding offer an intriguing coding logic to design competitive low-complexity encoding solutions. DSC refers to the separate encoding but joint decoding of multiple correlated sources and is supported by the information theoretic foundations laid by Slepian and Wolf [22]. Although [22] considered a lossless coding scenario, Wyner and Ziv [26] extended the latter findings by adding a quantization step, hence facilitating distributed lossy compression, referred to as Wyner-Ziv (WZ) coding. Video coding systems designed according to the principles of Wyner-Ziv theory, known as distributed video coding (DVC) or WZ video coding systems, no longer designate the encoder as the sole responsible for exploiting the spatio-temporal correlation but rather allow for sharing this responsibility with the decoder.

Coincidentally, the emergence of low-cost wireless multimedia sensor networks and processing technology [18] has attracted the attention of both the academic world and industry [27]. Wireless video sensors, in combination with tracking and detection algorithms [7], monitoring specific scenes, can provide security and surveillance. In this context, DVC has been identified as a promising enabling technology for low-power mobile media applications [20,27].

In practical DVC, an original source signal $X$ is coded independently using powerful channel codes. The decoder in turn first generates a prediction of the signal $X$, called the side information (SI) signal $Y$, which is subsequently used in the channel decoding of $X$. To this end, an accurate knowledge of the virtual correlation channel, capturing the conditional dependencies between $X$ and its SI is of paramount importance for efficient decoding.

In this regard, we have introduced the so-called SI dependent (SID) additive correlation channel model.
In particular, an SID channel assumes the additive noise component \( N \) as distributed according to a zero-mean Laplacian with standard-deviation \( \sigma(y) \), which varies depending on the actual realization \( y \) of the SI \( Y \). It has been demonstrated that the Laplacian SID correlation channel model describes the actual correlation channel more accurately than the conventional SI independent (SII) channel model [12]. Additionally, we have introduced an efficient online SID transform domain correlation channel estimator in [10], which is not confined to a particular WZ architecture.

Producing accurate SI at the decoder is a critical factor for the compression performance of a WZ codec. Motion-compensated interpolation (MCI) based SI generation [3,4] interpolates already decoded frames along the estimated motion field based on a linear translational motion model. However, the accuracy of a linear motion model degrades when the motion becomes highly irregular or the distance to the reference frames increases [16]. In such circumstances, the quality of the resulting SI dwindles and demands a large amount of channel coding rate in order to be corrected, hence reducing compression efficiency.

As a countermeasure, auxiliary information, called hash information, can be sent to the decoder to assist motion estimation, resulting in more accurate SI, even under strenuous conditions [1,6]. In this context, we have introduced a hash-based video coding architecture called spatial domain unidirectional DVC (SDUDVC) in [13]. Coarsely quantized versions of the original non-key frames served as a hash, based on which overlapping block motion estimation (OBME) and probabilistic motion compensation generated the reconstructed frames at the decoder. Lacking true WZ channel coding, the system of [13] evades the use of a feedback channel. Our hash-based video coding scheme presented in [23] not only added a WZ layer in the transform domain to [13], but in addition, it featured a novel probabilistic motion compensation technique, which combines the temporal predictors from the different overlapping blocks by means of weighted averaging, where the weights are calculated based on the combined knowledge of the hash and the estimated correlation statistics.

Our hash-based architectures introduced in [11,14] employed a down-sampled and H.264/AVC intra coded version of the original WZ frames as a hash. The hash-based OBME algorithm was adapted as to accommodate the alternative hash and the SI values were generated as the average of the candidate temporal predictors. In addition, [11] refined the quality of the SI during decoding to further boost the compression performance.

This work introduces a novel multi-hypothesis probabilistic motion compensation technique in a hash-based WZ architecture, which combines the large collection of temporal predictors for each pixel, generated through OBME, into a single value. The employed hash comprises a downsampled version of the hash utilized in [13,23]. The new multi-hypothesis maximum likelihood motion compensation (MLMC) technique requires knowledge of the conditional dependency between the temporal predictors and the original source values in order to generate accurate approximations of the original frames at the decoder. To this end, the correlation estimation algorithm from [10] is properly modified to use information from the hash and operate in the spatial domain.

The remainder of the paper is structured as follows. Section 2 provides a detailed description of our DVC scheme. Section 3 offers an in-depth explanation of the novel MLMC approach. Experimental results for the proposed DVC scheme, comparing it against a set of reference codecs, are provided in Section 4. Section 5 concludes the paper.

2. System architecture

2.1. The encoding procedure

A graphical overview of the proposed transform-domain WZ video codec is depicted in Fig. 1. The input sequence is split into key frames \( I \) and WZ frames \( X \), which are organized into groups of picture (GOPs). The key frames are coded with H.264/AVC [25], operating in intra frame mode only, while the WZ frames are coded according to the procedure presented in [2], that is, they undergo a DCT followed by quantization and the resulting quantization indices are coded with a Slepian-Wolf (SW) coder based on the rate-adaptive low-density parity-check accumulate (LDPCA) codes presented in [28].

For every WZ frame, the encoder creates hash information, which constitutes a crude representation of the original frame. The hash serves the same function as proposed in [13], namely as an aid to the decoder during motion estimation and SI generation. Figure 2 illustrates the hash formation and coding pipeline. The original WZ frames are downsampled by a factor \( \xi \) prior to coarse quantization. Similar to [13,23], uniform quantization with a quantization step-size \( q^{K-b} \) is ap-
applied, which is equivalent to retaining the $b$ most significant bitplanes of the original sample values having a bit-depth $K$. The obtained quantization indices are spatially decorrelated, entropy coded and sent to the decoder.

In particular, let $X$ denote the samples in the original WZ frame of width $W$ and height $H$. Let $s = (i, j)$ indicate a pixel position in the frame, where $i \in [0, W - 1]$ and $j \in [0, H - 1]$ are the column and row index, respectively. With this notation, $X(s)$ designates the sample at location $s$ in the original WZ frame.

Prior to quantization, the WZ frames are downscaled by a factor $\xi = 2^k$, $k \in \mathbb{N}$. Let $X'$ represent the samples in the downscaled frames with width $W' = W/\xi$ and height $H' = H/\xi$ and let $X'(s')$ represent the sample at position $s' = (i', j')$, where $i' \in [0, W' - 1]$ and $j' \in [0, H' - 1]$ are respectively the column and row in the downsampled frame. Downsampling is performed by decimation without anti-aliasing filtering, that is, the downsampling operation simply removes values from the original frame. In other words, the samples $X'(s')$ are obtained as $X'(s') = X(\xi s'), i' = 0, 1, \ldots, W', j' = 0, 1, \ldots, H'$. Subsequently, the samples $X'$ are quantized according to:

$$X'_b = \left\lfloor \frac{X'}{2^{K-b}} \right\rfloor$$

where $\lfloor \cdot \rfloor$ stands for taking the integer part, $K$ designates the bit-depth of the original samples and $b$ is the number of retained bitplanes. The resulting hash $X'_b$ is similar to a low-resolution grey-scale image containing $2^b$ distinct values, as illustrated in Fig. 2 for $b = 2$ and $\xi = 2$.

In order to remove spatial redundancy while maintaining limited hash encoding complexity, the quantization indices $X'_b$ are decorrelated using the edge-adaptive prediction scheme of JPEG-LS [24]. The resulting prediction errors are mapped to a new set of symbols in the range $[0, 2^b - 1]$ by modulo arithmetic. These symbols are then converted to sequences of binary symbols (bins) using unary coding and every symbol is subsequently entropy coded using binary arithmetic coding.

2.2. The decoding procedure

At the decoder side, the key frame and hash bitstream is decoded and the reconstructed key frames $\hat{I}$ and hash $X'_b$ are stored for future referencing. In order to enable LDPCA decoding of the WZ frames, the decoder creates SI in a two-stage process. First, the decoded hash is used to perform hash-based OBME, which generates a cluster of temporal predictors for each pixel in the WZ frame. In the second stage, a novel maximum likelihood motion compensation technique distils a single value from every cluster of predictors to serve as SI.

Before commencing motion estimation, the decoded hash $X'_b$ is upscaled to the original frame dimensions by upsampling via mere zero insertion. It is important to note that no interpolation filter is employed during upsampling. Instead, the zero-valued samples introduced by upsampling will be ignored during the block match-
The upsampled hash frame $\hat{X}_b^T$ is divided into overlapping blocks $\beta$ of size $B \times B$ pixels, with an overlap of $\varepsilon$ pixels. For every such block, the best matching block $\theta$, within search range $sr$, is identified in the reference frames.

The upsampled hash $\hat{X}_b^T$ contains the quantization indices representing the $b$ most significant bit-planes of the original WZ frames at pixel positions $s = \xi s'$, where $s' = (i', j')$, $i' = 0, 1, \ldots, W' - 1$, $j' = 0, 1, \ldots, H' - 1$. Hence, $\hat{X}_b^T$ contains reliable information solely at the pixel positions $\xi s'$. Since the reliable positions contain coarse quantization indices, traditional error metrics, like SAD or MSE, are inaccurate block matching criteria. Therefore, a more suitable matching error is applied.

Let $R_b$, $t \in \{0, 1\}$ represent the samples of either the past or future reference frame and let $\psi = (v_i, v_j)$, $-sr \leq v_i, v_j \leq sr$ be a motion vector within the search window defined by $sr$. The motion estimation algorithm chooses the best motion vector $\psi$ according to:

$$\psi = \arg \max_{\psi} \sum_{s \in \xi} \delta \left( \hat{X}_b^T(s) - \left[ \frac{R_b(s - \psi)}{2\kappa - b} \right] \right),$$

where $\delta(x)$ is the Kronecker delta function and $s$ is confined to all reliable pixel positions within a block. In other words, the motion estimation will maximize the number of equal quantization indices of collocated pixels in the blocks under consideration, where only reliable positions are taken into account.

In this way, for every pixel position $s$ in the WZ frame, OBME creates a set of temporal predictors $T_s = \{\psi_0, \psi_1, \ldots, \psi_{2^{\kappa} - 1}\}$. The elements $\psi_i$ correspond to the co-located pixels in the blocks $\theta_i$, which have been identified as the best match for the overlapping blocks $\beta_i$ covering the pixel position $s$.

The next step is to combine all the elements of $T_s$ into a single SI value, which is accomplished by means of a novel MLMC technique, explained in detail in Section 3. The resulting SI frame $Y$, is DCT transformed and acts as SI for the WZ decoding of $X$ in the transform domain. An online transform domain correlation channel estimation (TDCEC) is performed, according to the procedure introduced in [10]. In short, the algorithm determines the conditional probability mass function (PMF) of the quantization bins (defined by the already decoded bitplanes of the quantization indices of a coded DCT band) given the SI. From this intermediary PMF, the SID model parameters, that is, the shape parameter of the Laplacian distributions, given the SI, are derived. The current model approximation is then used to LDPC decode the next bitplane in the DCT frequency band after which the conditional PMF and SID model parameters are calculated anew.

When the WZ frames are decoded, they are optimally reconstructed, according to the procedure followed in [3, 11, 23], after which they undergo an inverse DCT. The reconstructed WZ frames inside a GOP remain stored in the reference frame buffer, where they serve as reference frames for the temporal prediction of WZ frames belonging to a lower temporal level [21].
3. Maximum likelihood motion compensation

3.1. Formulating the likelihood function

This section entails an in-depth presentation of the novel MLMC technique to generate the final SI values after the execution of OBME. Denote the discrete random variables representing the samples in the original WZ frame and the corresponding SI frame by \( X \) and \( Y \), respectively. Let \( T_s = \{ \psi_0, \psi_1, \ldots, \psi_{M-1} \} \) denote the set containing the candidate temporal predictors obtained by means of OBME for the WZ frame sample at location \( s \). MLMC is responsible for deducing a single value \( y \) from the set \( T_s \) to serve as SI at pixel position \( s \). With this goal, we propose the following likelihood function:

\[
L_s(x) = \prod_{i=0}^{M-1} p_{\psi|x}(\psi_i|x), \tag{3}
\]

where \( \Psi \) is a discrete random variable representing the temporal predictors and \( p_{\psi|x}(\psi_i|x) \) is the conditional PMF of \( \Psi \) conditioned on \( X \).

In order to maximize the likelihood function given by Eq. (3), the correlation between the candidate predictors and the original source \( X \) is first modelled as a conditional PMF. Since the temporal predictors \( \psi_i \) correspond to pixels in the reference frames, the conditional PMF between the original source \( X \) and the random variable \( \Psi \), that is, \( p_{X|\Psi}(x|\psi_i) \), is approximated by the conditional PMF \( p_{X|R_t}(x|r_t) \) between \( X \) and the source of candidate predictors, namely, the reference frames \( R_t \), \( t \in \{0,1\} \). Notice that estimating \( p_{X|R_t}(x|r_t) \) is conceptually different from TDC-CE [10], which is estimating the virtual correlation channel for WZ decoding [10]. Specifically, the TDCCE generates the model parameters for the conditional dependencies describing the virtual correlation channel between the original source \( X \) and the SI in the transform domain. This means that TDCCE is performed after motion compensated prediction at the decoder and is used to feed the SW decoder with the necessary soft-input information. On the contrary, the correlation estimated in the context of our MLMC (i) is performed prior to motion compensation, (ii) is set in the spatial domain, and (iii) serves as a means to obtain the statistical information required for MLMC.

3.2. Instantiating the correlation model

Denote by \( R_0 \), \( R_1 \) the discrete random variables representing the samples in the reconstructed past and future reference frames, respectively, while \( r_t, t \in \{0,1\} \) designates the realizations in one of the reference frames, taking on values within the range, \( [0,2^K-1] \), where \( K \) is the bit-depth of the original frame samples. In order to estimate \( p_{X|R_t}(x|r_t) \), we start by modelling the correlation \( f_{X|R_t}(\tilde{x}|r_t) \) between the samples in the original WZ frame and the reference frames \( R_t \). Inspired by [12], \( f_{X|R_t}(\tilde{x}|r_t) \) is modelled by a conditional PDF, which is assumed to be Laplacian distributed centred on \( r_t \) with standard deviation \( \sigma(r_t) \) and where the continuous random variable \( \tilde{X} \) represents the samples in the original WZ frame. In other words:

\[
f_{\tilde{X}|R_t}(\tilde{x}|r_t) = \frac{1}{\sqrt{2\pi\sigma(r_t)}}e^{-\frac{(\tilde{x}-r_t)^2}{2\sigma^2(r_t)}}. \tag{4}
\]

The parameters \( \sigma(r_t) \) can be estimated online similar to the methodology postulated in [10]. We note that the algorithm presented in [10] is set in the transform domain where the model parameters were successively refined during the SW decoding of the bitplanes of the quantization indices of the different frequency bands. Therefore, the algorithm has to be adapted to the current setting for which the correlation needs to be estimated in the spatial domain. Moreover, in contrast to [10], there is no opportunity for refining the correlation model.

Extrapolating the principles of [10] to the pixel domain, the PDF \( f_{X|R_t}(\tilde{x}|r_t) \) could be estimated from the conditional PMF \( p_{X|R_t}(x|r_t) \), where the discrete random variable \( X_r \) represents a coarsely quantized version of the original WZ frame.

However, such a coarsely quantized version of the original WZ frame is only partially available to the decoder. In fact, the hash is the sole available reliable source at the decoder based on which one can gather statistics on the original WZ frame. Bear in mind that the hash is created by downsampling the original WZ frame by a factor \( \xi = 2^k, k \in \mathbb{N} \) followed by coarse quantization retaining the \( b \) most significant bitplanes. That is, the hash contains values in the range \( [0,2^b-1] \). Therefore, one can use the hash together with each of the reference frames to obtain an estimate of \( p_{X|R_t}(x|r_t) \), as explained below.

Notice that no viable upscaling method for the hash exists that can generate trustworthy quantization index samples at the missing pixel positions. Therefore, in our technique the reference frames \( R_t, t \in \{0,1\} \) are first downsampled by a factor \( \xi \), according to the same method as during the creation of the hash, yielding the downsampled frames \( R'_t \). Then the correlation between the hash \( X'_r \) and downsampled reference frames \( R'_t \) is estimated. As a remark, observe that the downsam-
pling factor $\xi$ influences the statistical support for the correlation estimation.

In the next step of our algorithm, the joint PMF $p_{X|B}(x_b^s, r_t^s)$ is calculated by normalizing the joint histogram, counting the occurrences of co-located pairs $(x_b^s, r_t^s)$, where $x_b^s$ and $r_t^s$ take on values in the range $[0, 2^b - 1]$ and $[0, 2^K - 1]$, respectively. Then, the conditional PMFs $p_{X|R_t}(x_b^s | r_t^s)$, are computed by dividing the joint PMF by the empirical marginal PMF $p_{R_t}(r_t^s)$. The PMF $p_{X|R_t}(x_b^s | r_t^s)$ serves as an approximation for the PMF corresponding to the non-downsampled versions of the frames, i.e., $p_{X_b|R_t}(x_b | r_t)$.

We note that, due to downsampling, a particular realization $r_t$ might have no equivalent $r_t'$ in the downsampled reference frames $R_t'$. Hence, no empirical PMF conditioned on that particular realization of $R_t$ can be derived. This is handled by assigning a maximal level of uncertainty on $X_b$ to unobserved realizations of $R_t$, that is, a uniform distribution over all $2^b$ possible quantization indices in $X_b$.

Formally, the estimation of $p_{X_b|R_t}(x_b | r_t)$ is given by:

$$p_{X_b|R_t}(x_b | r_t) =
\begin{cases}
p_{X_b|R_t}(x_b' | r_t') & \text{if } r_t' \in R_t \\
2^{-b} & \text{otherwise}
\end{cases},$$

with:

$$\sum_{x_b=0}^{2^b-1} p_{X_b|R_t}(x_b | r_t) = 1, \quad \forall r_t \in [0, \ldots, 2^K - 1].$$

Similarly, a specific realization $x_b$ in $X_b$ might have been removed from the hash $X_b'$ due to the downsampling process. However, as $x_b \in [0, \ldots, 2^b - 1]$ has a limited range of possible values due to coarse quantization (e.g. $b = 2$), the absence of a particular realization $x_b$ is assumed to be accredited to its non-occurrence in $X_b$ rather than being deleted during downsampling.

Once the empirical conditional PMF $p_{X_b|R_t}(x_b | r_t)$ is determined, the standard-deviation $\sigma(r_t)$ of the particular Laplacian distribution $f_{\tilde{X}|R_t}(\tilde{x}|r_t)$ corresponding to the PMF $p_{X_b|R_t}(x_b | r_t)$ is found by solving [10]:

$$\int_{q_l(x_b)}^{q_u(x_b)} f_{\tilde{X}|R_t}(\tilde{x}|r_t) d\tilde{x} = 0 \quad (7)$$

where $q_l(x_b)$ and $q_u(x_b)$ are the respective lower and upper bound of the quantization interval defined by quantization index $x_b$. The solution, along with the proof of existence and uniqueness, can be found in [10].

### 3.3. Performing motion compensation

By assigning every predictor $\psi_i$ to its equal $r_i$ value in the corresponding reference frame, the conditional distribution $f_{\tilde{X}|\psi}(\tilde{x}|\psi_i)$ for every $\psi_i$ is equal to the corresponding $f_{\tilde{X}|R_t}(\tilde{x}|r_t)$, with $\sigma(\psi_i) = \sigma(r_t)$, where the standard-deviation $\sigma(r_t)$ was found by solving Eq. (7). Note that these correlation models are continuous. However, in reality pixel values are discrete, taking on integer values in the range $[0, 2^K - 1]$. Therefore, we derive the conditional PMF $p_{X|\psi}(x|\psi_i)$ by integrating $f_{\tilde{X}|\psi}(\tilde{x}|\psi_i)$ over a unit interval centred on the integer values within the range $[0, 2^K - 1]$:

$$p_{X|\psi}(x|\psi_i) = \int_{m(x)}^{n(x)} f_{\tilde{X}|\psi}(\tilde{x}|\psi_i) d\tilde{x},$$

where $x = 0, 1, \ldots, 2^K - 1$ and the integration bounds are given by Eqs (9) and (10).

$$m(x) = \left\{ \begin{array}{ll} x - 0.5 & \text{if } x > 0 \\ -\infty & \text{if } x = 0 \end{array} \right.$$ (9)

$$n(x) = \left\{ \begin{array}{ll} x + 0.5 & \text{if } x < 2^K - 1 \\ +\infty & \text{if } x = 2^K - 1 \end{array} \right.$$ (10)

Invoking Bayes’ law, the value $x_{ML}$ maximizing the likelihood function $L_{\psi}(x)$, in Eq. (3), can be found as:

$$x_{ML} = \arg \max_x \prod_{i=0}^{M-1} \frac{p_{X|\psi}(x|\psi_i) p_{\psi}(\psi_i)}{p_X(x)} \quad (11)$$

where $p_X(x)$ and $p_{\psi}(\psi_i)$ are prior PMFs. The posterior PMF $p_{X|\psi}(x|\psi_i)$ is given by Eq. (8).

It is clear from observing Eq. (11) that the factors $p_{\psi}(\psi_i)$ do not depend on $x$ and therefore do not affect the maximization process. Assuming the decoder has no prior information on the random variable $X$, all realizations $x$ are assumed to be equally likely, thereby rendering $p_X(x)$ of no influence on the maximization process. Integrating these observations and assumptions in Eq. (11) yields,

$$x_{ML} = \arg \max_x \prod_{i=0}^{M-1} p_{X|\psi}(x|\psi_i) \quad (12)$$

Finally, letting $x_{ML}$ obtained for position $s$ in the WZ frame serve as SI value for that particular location yields $y = x_{ML}$. 

---

*F. Verbist et al. / Maximum likelihood motion compensation for distributed video coding*
4. Experimental results

4.1. Compression performance

The performance of the proposed transform domain WZ video coding scheme was compared against a germane collection of state-of-the-art video compression systems including efficient DVC systems as well as low-complex representatives of conventional video codecs.

The first system, the DISCOVER codec [3], is a well established reference, delivering state-of-the-art compression performance in DVC. Similar to the proposed scheme, DISCOVER applies WZ coding in the transform domain but generates SI based on the latest MCI techniques [4,5]. The second reference codec is our spatial-domain unidirectional DVC (SDUDVC) scheme proposed in [13]. SDUDVC encodes the key frames of every GOP using H.264/AVC intra frame coding. The remaining frames in a GOP undergo coarse quantization forming a hash which is coded with a basic entropy codec. At the decoder, hash-based OBME produces the reconstructed frames directly in the spatial domain. SDUDVC delivers respectable compression performance without a transform domain WZ codec or a feedback channel. In addition, we have included a comparison against several of our previous WZ architectures [11,23], all employing a WZ layer in the transform domain and advancing significantly over SDUDVC in terms of compression performance.

Besides these DVC systems, two low-complexity conventional coding schemes are retained as a reference. Both schemes are configurations of the state-of-the-art H.264/AVC codec and avoid computationally expensive motion estimation at the encoder. The first version, H.264/AVC intra, encodes each frame in a video sequence separately, solely exploiting the intraframe spatial redundancies. H.264/AVC intra frame coding combines multi-hypothesis directional intra prediction followed by discrete cosine transformation of the prediction residuals with advanced context-based entropy coding, and is considered to be one of the most efficient intra frame coding schemes. H.264/AVC intra constitutes a well established benchmark when evaluating DVC solutions [3,6,9,19].

In a second configuration, additional prediction features of the H.264/AVC coding standard are activated. Apart from exploiting the spatial correlation, H.264/AVC no motion also removes temporal redundancies by means of simple differential coding principles. H.264/AVC no motion generally outperforms H.264/AVC intra frame coding at the cost of slightly increased encoder complexity. However, it needs emphasizing that both H.264/AVC configurations are significantly more complex than the assessed DVC solutions.

With regard to the evaluation of the proposed DVC scheme compared to H.264/AVC intra, H.264/AVC no motion and DISCOVER, experimental results are reported on the complete Foreman, Soccer, Carphone and Football sequences at QCIF resolution, a frame rate of 15 Hz and for a GOP size of 2, 4 and 8. Additionally, the results from our SDUDVC [13] on Foreman and Soccer have been included as well.

Concerning the configuration of the proposed DVC, the OBME module was configured with an overlap size $\varepsilon = 2$ and a block size of $B = 16$. The motion search was executed in an exhaustive manner within a search range of $\pm 15$ pixels at integer-pel accuracy. The hash was generated from the original WZ frames, which were downsampled by a factor $\xi = 2$ prior to quantization. A total number of $b = 2$ bit-planes were retained in the final hash. Both parameters $\xi$ and $b$ were chosen to obtain a balance between the accuracy of the motion estimation, the reliability of the spatial correlation estimation and the required rate to code the hash. Smaller $b$ and larger $\xi$ may result in a lower hash rate but lead to a poorer quality of the SI, since the motion estimation accuracy drops and the spatial domain correlation estimation algorithm no longer has the statistical support to generate faithful estimates. The MLMC was implemented numerically where for every pixelposition $s$ the search space was shrunk to $[\mu_{ss} - \sigma_{ss}/2, \mu_{ss} + \sigma_{ss}/2]$, with $\mu_{ss}$, $\sigma_{ss}$ denoting the respective mean and standard deviation of the appropriate set of temporal predictor pixels $T_{ss} = \{\psi_0, \psi_1, \ldots, \psi_{M-1}\}$. Figure 3 shows the compression performance of the proposed DVC with respect to the reference codecs for the Foreman (a–c) and Soccer (d–f) sequences. The Foreman sequence exhibits a reasonable amount of motion activity as well as intricate facial movements while severe camera panning causes a complete scene change towards the end of the sequence. Comparing the compression performance of the different DVC systems for this sequence, the proposed DVC indisputably exhibits the best performance with Bjøntegaard rate savings [8] of up to 25% with respect to DISCOVER in GOP of 8. The hash-based SDUDVC fails to reach the efficiency level of both competitor DVC schemes. When comparing the DVC solutions against the traditional predictive H.264/AVC codec, the experimental results indi-
cate that, for a GOP of 2, H.264/AVC no motion is the superior system in terms of compression efficiency. The performance of H.264/AVC intra is considerably lower, operating approximately on par with the proposed DVC and DISCOVER. When the GOP size is increased to 4 – (c) – H.264/AVC no motion is no longer able to take advantage of the temporal correlation by means of low-complexity motion compensation measures but has to resort primarily to intra prediction, which roughly equalizes its performance with H.264/AVC intra. In some cases, H.264/AVC intra even slightly outperforms H.264/AVC no motion. This seems counterintuitive but is the result of the fact that intra macroblocks in B-slices cost more to code than in I-slices due to slight differences in the entropy coding [17]. The convergence of H.264/AVC no motion to H.264/AVC intra with growing GOP size is regularly observed.
The Soccer sequence contains a fragment of a soccer game recorded at medium to short range, resulting in rather complex motion content with a wide range of accelerations and frequent camera panning. It is well known that under such circumstances, WZ based systems are hard-pressed to create accurate SI and suffer a notable performance loss compared to both H.264/AVC intra and H.264/AVC no motion, as confirmed by the experimental results. The rate-distortion behaviour of the relevant codecs for the Soccer sequence is presented in Fig. 3(d–f) for a GOP of respectively 2, 4 and 8. The proposed DVC solution manages to notably reduce the performance gap between classical video coding systems and DVC systems, particularly when the GOP size is large.

Figure 4 shows the compression performance obtained on the Carphone (a–c) and Football (d–f) sequences. Carphone is characterised by complex fa-
Table 1
Bjøntegaard compression gains of the proposed DVC with respect to DISCOVER

<table>
<thead>
<tr>
<th>Sequence</th>
<th>GOP2</th>
<th></th>
<th>GOP4</th>
<th></th>
<th>GOP8</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>∆R (%)</td>
<td>∆PSNR (dB)</td>
<td>∆R (%)</td>
<td>∆PSNR (dB)</td>
<td>∆R (%)</td>
<td>∆PSNR (dB)</td>
</tr>
<tr>
<td>Foreman</td>
<td>0.07</td>
<td>0.02</td>
<td>13.94</td>
<td>0.87</td>
<td>24.98</td>
<td>1.66</td>
</tr>
<tr>
<td>Soccer</td>
<td>7.30</td>
<td>0.41</td>
<td>14.83</td>
<td>0.88</td>
<td>21.97</td>
<td>1.36</td>
</tr>
<tr>
<td>Carphone</td>
<td>2.45</td>
<td>0.19</td>
<td>6.45</td>
<td>0.41</td>
<td>13.99</td>
<td>0.83</td>
</tr>
<tr>
<td>Football</td>
<td>5.78</td>
<td>0.36</td>
<td>10.20</td>
<td>0.62</td>
<td>13.68</td>
<td>0.83</td>
</tr>
</tbody>
</table>

Table 2
Bjøntegaard compression gains of the proposed DVC compared to our previous hash-based DVC introduced in [23].

Negative values constitute a performance loss

<table>
<thead>
<tr>
<th>Sequence</th>
<th>GOP2</th>
<th></th>
<th>GOP4</th>
<th></th>
<th>GOP8</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>∆R (%)</td>
<td>∆PSNR (dB)</td>
<td>∆R (%)</td>
<td>∆PSNR (dB)</td>
<td>∆R (%)</td>
<td>∆PSNR (dB)</td>
</tr>
<tr>
<td>Foreman</td>
<td>-4.49</td>
<td>-0.33</td>
<td>5.93</td>
<td>0.34</td>
<td>8.93</td>
<td>0.53</td>
</tr>
<tr>
<td>Carphone</td>
<td>-4.10</td>
<td>-0.28</td>
<td>5.37</td>
<td>0.31</td>
<td>6.87</td>
<td>0.39</td>
</tr>
</tbody>
</table>

Table 3
Bjøntegaard compression gains of the proposed DVC with respect to our previous hash-based WZ architecture, presented in [14]

<table>
<thead>
<tr>
<th>Sequence</th>
<th>GOP2</th>
<th></th>
<th>GOP4</th>
<th></th>
<th>GOP8</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>∆R (%)</td>
<td>∆PSNR (dB)</td>
<td>∆R (%)</td>
<td>∆PSNR (dB)</td>
<td>∆R (%)</td>
<td>∆PSNR (dB)</td>
</tr>
<tr>
<td>Foreman</td>
<td>2.09</td>
<td>0.13</td>
<td>2.42</td>
<td>0.14</td>
<td>0.35</td>
<td>0.01</td>
</tr>
<tr>
<td>Carphone</td>
<td>4.77</td>
<td>0.32</td>
<td>6.67</td>
<td>0.41</td>
<td>5.92</td>
<td>0.35</td>
</tr>
</tbody>
</table>

cial gestures and contains additional high motion restricted to the window area of the fast-moving car. The camera remains fixed throughout the entire sequence. H.264/AVC no motion outperforms the competition in terms of compression efficiency for all GOPs, while the compression performance of H.264/AVC intra is considerably less. In fact, both the proposed DVC and DISCOVER even outperform H.264/AVC intra at low rates. In the medium rate region, the proposed WZ system maintains a performance similar to H.264/AVC intra, slightly losing ground as the rate increases further. In comparison to DISCOVER, the proposed DVC system attains Bjøntegaard rate savings [8] of up to 13.99% for a GOP size 8.

Football is considered a hard-to-code sequence with a very high degree of motion as well as complex camera movements as it trails several players during an offensive manoeuvre. Similar to the Soccer sequence, the characteristics of Football highly favour conventional H.264/AVC coding. Regarding the comparison between the different DVC systems, the proposed WZ codec consistently outperforms DISCOVER for all GOP sizes, with Bjøntegaard rate savings [8] of up to 13.68% in a GOP of 8.

Table 1 provides a comprehensive overview of the compression performance comparison between the proposed DVC and the state-of-the-art DISCOVER codec in terms of Bjøntegaard rate savings (%) and distortion reduction (dB) [8]. Table 1 clearly illustrates the proposed DVC’s superior performance, in particular when the GOP size grows.

In the following, we compare the proposed WZ architecture with our previous hash-based DVC schemes. The first system, introduced in [23], employs a hash identical to one used in the SDUDVC [13]. The SI is generated via OBME followed by probabilistic motion compensation, comprising advanced weighted averaging of the candidate predictors.

Table 2 shows the compression performance of the proposed DVC, compared to the architecture from [23], for the Foreman and Carphone sequence in a GOP of 2, 4 and 8. The results show that the proposed DVC suffers a performance loss in a GOP of 2 with Bjøntegaard [8] rate losses of 4.49% and 4.10% for Foreman and Carphone, respectively. However, the proposed DVC recovers for the larger GOP sizes, consistently outperforming [23]. The behaviour is explained by the nature of the hash in [23], which is not a downsampled version of the bitplane hash from the SDUDVC, as used in the proposed system. For [23] this (i) results in more accurate motion estimation and (ii) allows for coding only the difference between the hash the original frame in a WZ manner. However, the bitrate overhead to code the hash is roughly $\xi^2$ times higher than the rate of the downsampled version, for a downsampling factor $\xi$. Such a rate overhead undermines the performance
in the rate-distortion sense, in particular when the GOP size grows.

The second hash-based DVC is our architecture presented in [14], where the hash consists of a downscaled and low quality H.264/AVC intra coded version of the WZ frames. The SI is created by means of OBME and the obtained candidate temporal predictors are merely averaged. Additionally, since the nature of the hash in [14] is different, the hash is truly upsampled at the decoder using a Lanczos interpolation filter [15] and the sum of the absolute differences (SAD) error metric is used during OBME. This architecture was expanded in [11] by adding a SI refinement loop at the decoder. After the SW decoding of all the bitplanes of the quantization indices in the DC band, the partially decoded WZ frame is transformed back to spatial domain and the SI generation process is executed anew, yielding an improved SI frame. Adding SI refinement at the decoder significantly boosts the compression performance [11]. However, since the proposed DVC in this work does not feature any SI refinement at the encoder, we have only included the experimental results for the core architecture of [14] without SI refinement. Table 3 shows the compression gains on account of the proposed DVC with respect to [14], obtained on Foreman and Carphone. The results corroborate that the proposed scheme consistently outperforms [14] for all GOPs.

4.2. Complexity evaluation

It needs strong emphasizing that the superior compression performance of H.264/AVC comes at a price. Although H.264/AVC intra, and to a lesser extent H.264/AVC no motion, are considered to be lightweight encoding configurations of the H.264/AVC standard, their computational complexity is a great deal higher than that of the evaluated DVC systems.

To endorse this statement, the encoder complexity of the proposed DVC is analysed according to the methodology followed in [9,19], where encoder execution times were compared. In this work, time measurements were carried out on a x86 machine with a Intel(R) Core(TM) i7 CPU running at 2.20 GHz and with 16 GB of RAM. Our DVC was written in C++, compiled with Microsoft Visual Studio 2008 and running in release mode under Windows 7. We measured the execution times to encode the entire Foreman and Carphone sequence, at QCIF resolution and a frame rate of 15Hz, using H.264/AVC intra and the proposed DVC. The quality parameter (QP), controlling the quantization level of the H.264/AVC encoder was identical for the H.264/AVC intra coded sequences and the key frames of the WZ codec. The employed quantization matrices (QM), used to quantize the WZ frames in the transform domain, are equal to the QMs introduced in [2].

Table 4 shows the executing time of the proposed DVC, broken down in its prime components, that is, the encoding of the key frames, the LDPCA encoding of the WZ frames and dealing with the hash. As seen in Table 4, the majority part of the encoding complexity of the proposed DVC scheme is due to the H.264/AVC intra coding of the key frames. In contrast to MCI-based DVC systems, a hash-based WZ codec allocates additional resources at the encoder to code the hash information. However, the added complexity regarding the creation and coding of the hash, comprising quantization, prediction, binarization and entropy coding, is fairly modest with respect the combined resources required for key frame and WZ encoding.

For comparison, Table 5(a) contains the execution time for encoding Foreman and Carphone with H.264/AVC intra. Table 5(b) summarizes the execution time saved by adopting our hash-based DVC compared to H.264/AVC intra coding, displaying the ratio (%) of the total execution time of the proposed WZ scheme and H.264/AVC intra. It is clear that our WZ system brings significant savings, in particular for the larger GOP sizes, comprising around 30% of the execution time of H.264/AVC intra in a GOP of 8.
cies between the temporal predictors and the original video. In order to facilitate MLMC, the spatial domain conditional dependencies between the temporal predictors and the original source are estimated online using an adaptation of our previous algorithm to estimate the transform domain SID correlation channel statistics [10].

Experimental results testify to the state-of-the-art distributed coding performance of the proposed WZ video coding architecture. Using a downsamped and coarsely quantized version of the original WZ frames as a hash, the proposed DVC generates a large collection of temporal predictors by means of OBME from which a new multi-hypothesis probabilistic motion compensation technique generates accurate SI. In order to facilitate MLMC, the spatial domain conditional dependencies between the temporal predictors and the original source are estimated online using an adaptation of our previous algorithm to estimate the transform domain SID correlation channel statistics [10].

5. Conclusions

This paper proposed a novel MLMC scheme in a hash-based transform domain WZ video coding architecture. Using a downsamped and coarsely quantized version of the original WZ frames as a hash, the proposed DVC generates a large collection of temporal predictors by means of OBME from which a new multi-hypothesis probabilistic motion compensation technique generates accurate SI. In order to facilitate MLMC, the spatial domain conditional dependencies between the temporal predictors and the original source are estimated online using an adaptation of our previous algorithm to estimate the transform domain SID correlation channel statistics [10].

Experimental results testify to the state-of-the-art distributed coding performance of the proposed WZ video coding architecture featuring the presented MLMC. The proposed system brings significant gains over the reference DISCOVER codec, especially for sequences containing complex motion patterns or when the distance to the reference frames used for temporal prediction is large. The proposed scheme considerably advances over our previous hash-based DVC solutions and further diminishes the existing performance gap between state-of-the-art transform domain WZ video codecs and traditional low-complexity predictive coding solutions. Taking full advantage of the potential of Wyner and Ziv’s philosophy, the encoding complexity of the proposed systems is reduced to up to 30% of the complexity of the benchmark H.264/AVC intra codec.

References


