Adaptive Correlation Estimation With Particle Filtering For Distributed Video Coding

Shuang Wang*, Lijuan Cui*, Lina Stanković†, Vladimir Stanković†, and Samuel Cheng*
*School of Electrical and Computer Engineering
University of Oklahoma-Tulsa, Tulsa, OK, 74135
Email: {shuangwang, lj.cui, samuel.cheng}@ou.edu
†Dept. of Electronic and Electrical Engineering
University of Strathclyde, Glasgow, UK
Email:{lina.stankovic, vladimir.stankovic}@eee.strath.ac.uk

Abstract

Distributed Video Coding (DVC) is rapidly gaining popularity as a low cost, robust video coding solution, that reduces video encoding complexity. DVC is built on Distributed Source Coding (DSC) principles where correlation between sources to be compressed is exploited at the decoder side. In the case of DVC, a current frame available only at the encoder is estimated at the decoder with side information generated from other frames available at the decoder. One of the main challenges in DVC design is that correlation among the source and side information needs to be estimated online and as accurately as possible. Since correlation dynamically changes with the scene, in order to exploit the robustness of DSC code designs, we integrate particle filtering with standard belief propagation (BP) decoding for inference on one joint factor graph to estimate correlation among source and side information. Correlation estimation is performed online as it is carried out jointly with decoding of the graph-based DSC code. Moreover, we demonstrate our joint bit-plane decoding with adaptive correlation estimation schemes within state-of-the-art DVC systems, which are transform-domain based with a feedback channel for rate adaptation. Experimental results show that our proposed system gives a significant performance improvement compared to the benchmark state-of-the-art DISCOVER codec (including correlation estimation) and the case without dynamic particle filtering tracking, due to improved knowledge of timely correlation statistics via the combination of joint bit-plane decoding and particle-based BP tracking.

This work was supported in part by NSF under grant CCF 1117886.

I. Introduction

Distributed compression or Distributed Source Coding (DSC) refers to separate compression and joint decompression of multiple correlated sources. DSC started as an information-theoretical problem in the renowned 1973 paper of Slepian and Wolf [1]. Slepian and Wolf considered lossless separate compression of two discrete sources, and showed that, roughly speaking, there is no performance loss compared to joint compression as long as joint decompression is performed. This remarkable result triggered significant information-theoretical research resulting in many extensions.

In 1976, Wyner and Ziv [2] considered a lossy version (i.e., with a distortion constraint) of the non-asymmetric Slepian-Wolf problem, where one source is available at the decoder as side information. Wyner and Ziv showed that for a particular correlation where source and side information are jointly Gaussian, there is no performance loss due to the absence of side information at the encoder.

A possible realization of DSC via the use of channel codes was proposed 1974 in [3], but due to the lack of any potential application of DSC, work on code designs, i.e., how to code the sources to approach given bounds, started only a decade ago. The first practical design was reported in 1999 [4] followed by many improved solutions (see and references therein). The key beauty of the developed DSC designs is that conventional channel coding can be used for compression [5].

These theoretical advances and powerful code designs have recently paved the way towards practical applications. Distributed Video Coding (DVC) or Wyner-Ziv video coding, is one of the earliest and most advanced applications of DSC to date. DVC [6], [7] was proposed as a solution for emerging many-to-one setups, such as video surveillance with tiny cameras and cell-to-cell communications, where low encoding complexity is a must. It brought a paradigm shift from the conventional centralized video coding architecture, where encoding complexity is much higher than decoding complexity, typical of traditional television broadcasting.

In DVC, the DSC paradigm is used to avoid a computationally expensive temporal prediction loop at the encoder by exploiting inter-frame video correlation with Wyner-Ziv (WZ) coding. It relies on the fact that with WZ coding, side information is not needed at the encoder. WZ source coding is usually realized by quantization followed by Slepian-Wolf (SW) coding of quantization indices based on channel coding [5]. Quantization is used to tune rate-distortion performance, while the SW coder is essentially a conditional entropy coder. The WZ decoder will thus comprise an SW decoder, which makes use of side information to recover the coded information. The SW decoder is followed by a minimum-distortion reconstruction of the source using side information and the SW decoded quantization index [4], [5].

Practical WZ code design based on quantization plus conventional channel codes is possible since correlation between the source and side information is seen as a virtual communication channel which can be expressed in the form $X = Y + N$, where X is the source to be recovered defined as the sum of the side information Y and noise N. As long as this virtual channel can be modeled by some standard communication channel, e.g., Gaussian, channel codes can be effectively employed. Designs [5], [8] based on trellis-coded quantization followed by advanced channel coding come very close to the bounds for two jointly Gaussian sources.

However, similar to the information-theoretical DSC framework [1], [2], the state-of-the-art SW and WZ code designs based on turbo and low-density parity-check (LDPC) codes perform well only when correlation statistics between sources are stationary and known at the encoder and decoder. The problem of online statistical correlation estimation between the source and side information is particularly important in DVC, since the scene dynamically and unpredictably changes. Thus in DVC we have a problem of WZ coding with non-stationary correlation noise with unknown statistics. Most DVC designs so far (with few exceptions) usually simplify this problem by modeling correlation noise, i.e., the difference between the source and side information, as Laplacian random variables and estimate the distribution parameters either based on training sequences or previously decoded data, which imposes certain loss especially for high-motion sequences. Non-stationarity of the scene has been dealt mainly by estimating correlation noise (e.g., on the pixel or block level) from previously decoded data and different initial reliability is assigned to different pixels based on the amount of noise estimated both in pixel- and transform-domains [9], [10], [11], [12], [13].

We propose in this paper an efficient way of estimating correlation between X and Y by tightly incorporating the process within the SW decoder and augmenting the SW code factor graph to include correlation variable nodes with particles such that particle filtering is performed jointly with belief propagation over the factor graph during the SW decoding process. Thus, in contrast to previous work [9], [10], [11], [12], [13], conventional belief propagation-based SW decoding and correlation statistics estimation are considered jointly. The proposed correlation estimation design is tested on a transform-domain DVC [14] with a feedback channel, but with joint bit-plane coding [15].

This paper's contributions can thus be summarized as:

1) Introduce correlation noise estimation within the SW decoding process that takes into account both source and side-information statistics.
2) Construct a factor graph with adaptively connected regions incorporating joint bit-plane SW decoding and correlation variable nodes for correlation estimation to capture correlation statistics for

different video sequences.

3) Perform inference, applied over the augmented SW factor graph, using a modified Belief Propagation algorithm that works jointly with particle filtering on particles tied to the correlation variable nodes, to successively refine estimation of the correlation noise and source iteratively until convergence is reached. Statistics of the side information are inherently captured.
4) Incorporate the proposed joint bit-plane decoding as well as the adaptive correlation estimation design into a transform-domain state-of-the-art DVC.

We modified the state-of-the-art transform-domain DISCOVER [14] decoder by including both the proposed correlation-tracking algorithm and joint bit-plane decoding, comparing performance to that obtained with the traditional DISCOVER codec of [14] to demonstrate the effectiveness of the proposed approach. Moreover, by combining our proposed scheme with the estimation algorithm of [11], we demonstrate the refinement in correlation statistics introduced via our proposed scheme. Note that in the proposed DVC codec we did not introduce any changes in the encoding process with respect to classic DVC, maintaining low encoding complexity.

The paper is organized as follows. In the next section, we outline previous methods for correlation estimation. Factor graph construction is explained in Section III. We describe standard belief propagation (BP) and explain the concept of adaptive graph-based decoding incorporating particle filtering in Section IV. In Section V, the proposed scheme built within a state-of-the-art DVC setup is described, followed by experimental results in Section VI. The paper concludes with Section VII.

II. Background: Correlation Estimation

Despite advances in practical DSC designs, the widespread application of DSC is limited by the accurate estimation of correlation between source and side information. In this section, we review related work on the various approaches to correlation estimation for DSC in general, including DVC. We also highlight how our proposed correlation tracking algorithm differs from these approaches. A detailed treatment of our proposed algorithm including factor graph construction and message passing on the graph with particle filtering is provided in Sections III and IV.

In [16], [17], [18], [19], the correlation in the SW coding problem is modeled using a binary symmetric channel (BSC) with crossover probability as the parameter to be estimated. However, these correlation estimation studies in [16], [17], [18] assume that the correlation parameter is constant and does not change within the code block. To estimate non-stationary correlation statistics for the BSC correlation model during SW decoding, we proposed particle based belief propagation algorithm on augmented factor graph in our prior work [19]. Due to the close relationship between SW coding and channel coding, the proposed approach can also be used for channel state estimation [20]. However, these previous studies of correlation estimation in DSC focus on binary sources, which are not suitable for the non-binary frame sources in the DVC case.

In WZ coding, the encoder does not need to know/use side information, thus making it possible to accomplish predictive coding without encoder motion compensation. In a nutshell, a block of pixels/coefficients in the current frame is WZ encoded into a stream without any reference to previously encoded data, and the decoder uses all available information to generate a side information block that will be used to WZ decode the compressed stream. State-of-the-art WZ coding designs based on turbo, LDPC, and other graph-based codes are successfully used for DVC (see [21], [22], [23] and references therein).

Note that a key difference between conventional WZ coding and DVC is that in the former statistics of the correlation noise are known to both the encoder and decoder and does not change over time. In DVC, however, the decoder needs to generate side information using only the information it has; regardless of how side information is generated, correlation noise statistics will be unknown and dynamically change over time. Indeed, due to the non-stationarity of real scenes, WZ coding in DVC has to deal with varying correlation noise statistics.

Estimating correlation statistics has been identified as a key challenge in DVC. Usually, correlation error is modeled as a Laplacian random variable whose parameters are estimated from previously decoded frames. It has been shown however that these models are not accurate enough if there are occluded regions in the scene [9]. In this case, the correlation noise would be concentrated in the occluded areas, which are usually at the boundaries of moving objects.

In [10], the correlation noise is always modeled as Laplacian, but to capture the non-stationary nature of the scene, the correlation parameter was varied from pixel to pixel. The noise power is increased if the pixel difference between motion compensated blocks in the two key frames used to generate side information is high; otherwise, it is decreased. The reasoning behind this method is that if the difference between the two key frames is high then we have less confidence in their average and the noise variance is higher. Thus, incorporating this model within SW decoding ensures that the SW code assigns higher reliability to pixels that have been predicted with higher accuracy, that is, the difference between the key frames is smaller. Similarly, in [11], the Laplacian distribution is used with the parameter estimated online at the sequence, frame, block, and pixel level from decoded frames at the decoder. In [12], [13], improved online channel estimators are proposed that attempt to address the issue of difficulty in adaptive correlation in smaller spatial regions due to the difficulty in acquiring sufficient statistics. The above correlation models determine their parameters based on the noise realization in a given temporal or spatio-temporal neighbourhood. In [24], a side information dependent correlation noise model is proposed where the standard deviation of the Laplacian model is a function of a particular realization of the side information at each pixel position.

Note that in [9], [10], [11], [12], [13], [24], non-stationarity of a scene is addressed by changing the correlation model on-the-fly and supplying the SW decoder with different initial reliability estimates. However, all of these works assign each band or coefficient a Laplacian parameter, which is estimated using previous successfully decoded information, before the current band SW decoding, but do not refine the estimated Laplacian parameter using current decoded information when decoding is in process.

Since the SW decoding process refines starting beliefs, this paper postulates that unifying the process of correlation estimation and joint bit-plane decoding into a single joint process can provide better statistics estimate and consequently improved performance. Indeed, the performance improvement of such unification for pixel-domain DVC is demonstrated in our conference paper [25], while this paper will fully develop the solution for the transform-domain state-of-the-art DVC system and compare with benchmark DISCOVER codec [14] and correlation estimation of [11]. Note that this unification of correlation estimation and SW decoding will also enable the correlation estimator to take into account side information statistics and any of the methods of [9], [10], [11], [12], [13] can be used as an initial point that will be refined during SW decoding.

In this paper, unlike approaches described above, we introduce correlation noise estimation for non-binary sources within the SW decoding process of the DVC architecture, taking into account both source and side-information statistics. This is achieved by augmenting the SW code factor graph with correlation variable nodes for correlation estimation to adaptively capture correlation statistics for different video sequences. We model correlation noise with a Laplacian distribution, unlike previous approaches [16], [17], [18], [19] for DSC where correlation estimation was modelled by the BSC for binary sources. The non-binary nature of the correlation noise is addressed by incorporating particle filtering within the belief propagation inference algorithm applied over the overall graph connecting the correlation variables and SW factor graph, thus capturing source and side-information statistics at each iteration.

III. Factor graph construction

A factor graph is a particular type of graphical model that enables efficient computation of marginal distributions through the sum-product algorithm, commonly referred to as belief propagation. A factor

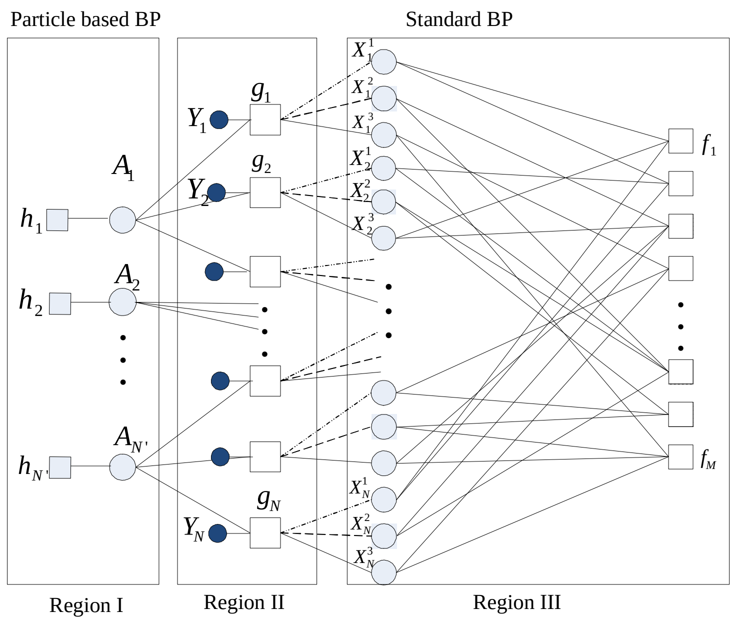

Fig. 1. Factor graph of joint bit-plane SW decoding with correlation estimation.

graph is characterized by a node (usually depicted by a circle) for every variable in the distribution, additional nodes (depicted by small squares) for each factor in the joint distribution and undirected links connecting each factor node to all of the variables nodes on which that factor depends. Factor graphs together with the sum-product algorithm are typically used for the decoding of capacity-approaching error-correcting channel codes, such as LDPC, which we employ for SW decoding.

We model the correlation between source and side information as Laplacian, and build a factor graph to capture joint bit-plane decoding, striving for efficient compression, accurate correlation modeling, and good performance. For WZ coding, we carry out joint bit-plane coding by first quantizing an N-length source sample x_i (pixel or transform coefficient in DVC), $i = 1, \cdots, N$, into $Q[x_i]$ using 2^{M_b} levels quantization (see Fig.1 for an example with $M_b = 3$). We denote $x_i^1, x_i^2, \cdots, x_i^{M_b}$ as the binary format of the index $Q[x_i]$, and denote $\mathbf{B} = x_1^1, x_1^2, \cdots, x_1^{M_b}, x_2^1, x_2^2, \cdots, x_N^{M_b}$ as the block which combines all the bit variables. The block $\mathbf{B}$ is encoded using LDPC-based SW codes and generates parity/syndrome bits $\mathbf{S} = s_1, s_2, \cdots, s_M$. This results in an $M_bN : M$ SW compression ratio.

In Region III, the factor nodes $f_1, f_2, \cdots, f_M$, connect the bit variable nodes X_i^j, $i = 1, \cdots, N$, $j = 1, \cdots, M_b$, and take into account constraints imposed by the received parity/syndrome bits. For the factor node $f_a, a = 1, \ldots, M$, we define the corresponding factor function

$$f_a(\tilde{\mathbf{x}}_\mathbf{a}, s_a) = \begin{cases} 1, & \text{if } s_a \oplus \bigoplus \tilde{\mathbf{x}}_\mathbf{a} = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

where $\tilde{\mathbf{x}}_\mathbf{a}$ denotes the set of neighbors of factor node f_a, and $\bigoplus \tilde{\mathbf{x}}_\mathbf{a}$ denotes the binary sum of all elements of the set $\tilde{\mathbf{x}}_\mathbf{a}$.

Let y_i, the variable of node Y_i, $i = 1, \cdots, N$, be i-th sample (pixel or transform coefficient in DVC) of the side information available at the decoder. To take into account the remaining correlation between quantization indices $Q[x_i]$ and side information y_i, we use a Laplacian distribution to model the statistical correlation by additionally defining correlation factor nodes g_i, $i = 1, \cdots, N$ in Region II. The factor function of g_i is defined as

$$g_i(Q[x_i], y_i, \alpha) = \int_{P(Q[x_i])}^{P(Q[x_i]+1)} \frac{\alpha}{2} e^{\alpha|x-y_i|} dx, \tag{2}$$

where α is the scale parameter of the Laplacian distribution, $P(\bullet)$ denotes the lower boundary of quantization partition at index "$\bullet$", i.e., if a coefficient x_i satisfies $P(\bullet) \leq x_i < P(\bullet+1)$, the quantization index $Q[x_i]$ of coefficient x_i is equal to "$\bullet$". In fact, the factor node g_i plays a role of providing a predetermined likelihood $p(y_i|Q[x_i], \alpha)$ to variable node X_i^j, $j = 1, \cdots, M_b$ for LDPC based SW decoding.

However, in practical applications such as DVC, it is rarely the case that correlation parameter α is assumed to be known *a priori*. We assume that α_t is unknown and varies over time, typical for correlated frames in a video sequence. Thus, it is necessary to perform online correlation parameter estimation to avoid the degradation of decoding performance. It also means that each factor node g_i will periodically update the likelihood $p(y_i|Q[x_i], \alpha_t)$ for the respective bit variable nodes $X_i^1, X_i^2, \cdots, X_i^{M_b}$ when a new estimate of correlation parameter α_t is available, instead of using a predetermined likelihood $p(y_i|Q[x_i], \alpha)$. To enable the online estimation of time-varying correlation parameter α_t, we introduce extra variable nodes A_l, and factor nodes h_l, $l = 1, 2, \ldots, N'$ (see Region I of Fig. 1). Here, we call the number of factor nodes in Region II connecting to each variable node A_l the connection ratio C [1], which is equal to three in the example in Fig. 1. In Region I, each variable node A_l is used to model the time-varying correlation parameter α_l of a block of C source samples. Moreover, the factor function $h_l(\alpha_l)$ of factor node h_l corresponds to the *a priori* distribution for variable α_l, which is taken as a uniform distribution. Consequently, by introducing correlation parameter estimation in Region I, likelihood factor

[1]To estimate a stationary correlation parameter, we can set the connection ratio equal to the code length. Moreover, connection ratio provides a trade-off between complexity and spatial variation. Non-stationary correlation parameter estimation and connection ratio are discussed in Section V.

function in (2) will be updated as

$$g_i(Q[x_i], y_i, \alpha_l) = \int_{P(Q[x_i])}^{P(Q[x_i]+1)} \frac{\alpha_l}{2} e^{\alpha_l |x - y_i|} dx. \tag{3}$$

The final factor graph capturing and connecting SW coding and correlation tracking (Fig. 1) comprises Region III, with a standard Tanner graph for bit-plane LDPC decoding, and Region I, with a bipartite graph capturing correlation parameter, and Region II, where factor nodes g_i, defined in (3) for Laplacian correlation connect the two regions.

IV. Message passing algorithm with particle filtering on constructed graph

In this section, we first review the standard belief propagation (BP) algorithm, and then describe the proposed message passing algorithm for efficient SW decoding and correlation estimation via the BP algorithm operating jointly with the particle filtering algorithm.

A. *Review of Belief Propagation Algorithm*

The BP algorithm is an approximate technique for computing marginal probabilities by exchanging the message between connected nodes. Denote $m_{a\to i}(x_i)$ as the message sent from a factor node a to a variable node i, and $m_{i\to a}(x_i)$ as the message sent from a variable node i to a factor node a. Loosely speaking, $m_{a\to i}(x_i)$ and $m_{i\to a}(x_i)$ can be interpreted as the beliefs of node i taking the value x_i transmitting from node a to i and from node i to a, respectively. The message updating rules can be expressed as follows:

$$m_{i\to a}(x_i) \propto \prod_{c\in\mathcal{N}^{\setminus a}(i)} m_{c\to i}(x_i) \tag{4}$$

and

$$m_{a\to i}(x_i) \propto \sum_{\mathbf{x}_a\setminus x_i} \left(f_a(\mathbf{x}_a) \prod_{j\in\mathcal{N}^{\setminus i}(a)} m_{j\to a}(x_j) \right), \tag{5}$$

where $\mathcal{N}^{\setminus a}(i)$ denotes the set of all neighbors' indices of node i excluding the index of node a; f_a is the factor function for factor node a; $\sum_{\mathbf{x}_a\setminus x_i}$ denotes a sum over all the variables in $\mathbf{x}_a$ that are arguments of f_a except x_i. Moreover, the BP algorithm approximates the belief of node i taking x_i as

$$b_i(x_i) \propto \prod_{a\in\mathcal{N}(i)} m_{a\to i}(x_i). \tag{6}$$

B. Particle based belief propagation algorithm

Based on the factor graph in Fig. 1, messages are passed iteratively between connected variable nodes and factor nodes in all the regions of the graph until the algorithm converges, i.e., the variable nodes converge to an estimate with the syndrome matching with the received one and we can stop updating their belief, or until a fixed number of iterations is reached. These messages (inferences or beliefs on source and correlation parameter) will represent the influence that one variable has on another. We group these types of messages into the three connected regions identified previously, thus generalizing BP. Hence, our correlation estimation exploits variations in side information in each bit-plane, and dynamically tracks spatial and temporal variations in correlation between source and side information.

Standard BP generally used for SW decoding, can handle only discrete variables. The correlation parameter, however, is not a discrete variable, since it varies continuously over time. Thus we only operate standard BP in Region III of the factor graph (Fig. 1), but cannot apply BP directly in Regions I and II for correlation parameter estimation. In Bayesian inference, the estimation of correlation parameter α_l corresponds to the estimation of its posterior distribution, i.e., $p(\alpha_l|\mathbf{y}_l)$, where $\mathbf{y}_l = (y_i|i \in \mathcal{N}^{\setminus h_l}(A_l))$, and $\mathcal{N}^{\setminus h_l}(A_l)$ represents the set of all neighbors' indices for a variable node A_l except the index of h_l. Then with the posterior distribution, the likelihood $p(y_i|Q[x_i], \alpha_l)$ is easily obtained to improve the LDPC-based SW decoding, where the likelihood $p(y_i|Q[x_i], \alpha_l)$ corresponds to the message $m_{g_i \to X_i^j}(x_i^j)$, $j = 1, 2, \cdots, M_b$. Based on Bayes' rule and LDPC-based SW coding, it is easy to verify that

$$\begin{aligned} p(\alpha_l|\mathbf{y}_l) &= \frac{\prod_{i \in \mathcal{N}^{\setminus h_l}(A_l)} p(\alpha_l) p(y_i|\alpha_l)}{\int_{\alpha_l} \prod_{i \in \mathcal{N}^{\setminus h_l}(A_l)} p(\alpha_l) p(y_i|\alpha_l)} \\ &= \frac{\prod_{i \in \mathcal{N}^{\setminus h_l}(A_l)} \int_{Q[x_i]} p(\alpha_l) p(Q[x_i]) p(y_i|Q[x_i]; \alpha_l)}{\int_{\alpha_l} \prod_{i \in \mathcal{N}^{\setminus h_l}(A_l)} \int_{Q[x_i]} p(\alpha_l) p(Q[x_i]) p(y_i|Q[x_i]; \alpha_l)}, \end{aligned} \tag{7}$$

where $p(\alpha_l)$, the *a priori* distribution for α_l, is modeled by the factor function $h_l(\alpha_l)$; $p(y_i|Q[x_i]; \alpha_l)$, the likelihood for y_i, is modeled by the factor function $g(y_i; Q[x_i], \alpha_l)$; $p(Q[x_i])$, the *a priori* distribution for quantized index of x_i, is captured by the message $m_{X_i^j \to g_i}(x_i^j)$ for each bit-plane defined in [26]. Then the posterior distribution (7) can be written as

$$\begin{aligned} p(\alpha_l|\mathbf{y}_l) &= \frac{1}{Z_l} h(\alpha_l) \prod_{i \in \mathcal{N}^{\setminus h_l}(A_l)} \sum_{\mathbf{x}_i^{M_b}} g(y_i; Q[x_i], \alpha_l) \prod_{j \in 1, 2, \cdots, M_b} m_{X_i^j \to g_i}(x_i^j) \\ &= \frac{1}{Z_l} m_{h_l \to A_l}(\alpha_l) \prod_{i \in \mathcal{N}^{\setminus h_l}(A_l)} m_{g_i \to A_l}(\alpha_l), \end{aligned} \tag{8}$$

where Z_l is a normalization constant, $\sum_{\mathbf{x}_i^{M_b}}$ denotes a sum over all the bit variables in $\mathbf{x}_i^{M_b}$, the value of message $m_{X_i^j \to g_i}(x_i^j)$ is updated iteratively by variable node X_i^j in Region III according to BP update rule (4), message $m_{h_l \to A_l}(\alpha_l) = h(\alpha_l)$ comes from prior factor node in Region I, and message

$m_{g_i \to A_l}(\alpha_l) = \sum_{\mathbf{x}_i^{M_b}} g(y_i; Q[x_i], \alpha_l) \prod_{j \in 1,2,\cdots,M_b} m_{X_i^j \to g_i}(x_i^j)$ comes from likelihood factor node in Region II according to the BP update rule (5).

However, since all the bit variables x_i^j, $j = 1, \cdots, M_b$, in $\mathbf{x}_i^{M_b}$ are discrete and take values 0 or 1, in (8), the message $m_{g_i \to A_l}(\alpha_l) = \sum_{\mathbf{x}_i^{M_b}} g(y_i; Q[x_i], \alpha_l) \prod_{j \in 1,2,\cdots,M_b} m_{X_i^j \to g_i}(x_i^j)$ has 2^{M_b} terms and the product of all the messages $\prod_{i \in \mathcal{N}^{\setminus h_l}(A_l)} m_{g_i \to A_l}(\alpha_l)$ is a mixture of $2^{M_b C}$ number of Laplacian distributions, where $C = |\mathcal{N}^{\setminus h_l}(A_l)|$ is the connection ratio, M_b is the number of bit-planes, and $M_b C$ can be a large number. Thus, the direct evaluation of the posterior distribution would be infeasible. To solve this problem, a possible way is to approximate the posterior distribution through particle filtering (PF) algorithm [27], which discretizes each continuous variable α_l in Region I with N_p number of particles $\alpha_l^1, \alpha_l^2, \cdots, \alpha_l^{N_p}$ sampled from distribution $p(\alpha_l|\mathbf{y}_l)$. The posterior distribution $p(\alpha_l|\mathbf{y}_l)$ corresponds to the belief of variable α_l in the LDPC-based SW coding shown in (8).

We therefore resort to integrating PF with the standard BP algorithm, which we refer to from now on as particle based BP (PBP) algorithm, in order to handle continuous variables, which mainly model each variable α_l with N_p particles $\alpha_l^1, \alpha_l^2, \cdots, \alpha_l^{N_p}$ in our problem. In PBP, the N_p sampled particles with associated weights over variable α_l simply correspond to N_p labels in standard BP, where the associated weight of each particle corresponds to the belief of each label. Then, both the value (i.e., location) of each label (i.e., particle) and the belief of each label will be updated after each iteration, which is different from standard BP that only updates the belief of each label after each iteration. Note that these changes do not affect the sum-product message update rules described in the standard BP algorithm in Section IV-A. Additionally, locations and corresponding weights of particles have to be adjusted over time. This is achieved by using systematic resampling [28] and Metropolis-Hastings (MH) [29] random walk perturbation after each message update.

Since we mentioned that we group these types of messages into the three connected regions identified previously, factor node g_i connecting Regions I and III plays an important role for providing periodically updated correlation parameter estimate for joint bit-plane SW decoding, and providing candidate decoded $Q[x_i]$ to refine the correlation parameter estimate. Thus, the iteratively updated likelihood $p(y_i|Q[x_i], \alpha_l)$ provided for LDPC-based joint bit-plane SW decoding that corresponds to the message for each bit plane

$m_{g_i \to X_i^j}(x_i^j)$, $j = 1, 2, \cdots, M_b$, can be written as

$$\begin{aligned} m_{g_i \to X_i^j}(x_i^j) &\propto \int_{\alpha_l} \sum_{\mathbf{x}_i^{M_b} \backslash x_i^j} g_i(y_i; Q[x_i], \alpha_l) m_{A_l \to g_i}(\alpha_l) \prod_{\tau \in \{1,2,\cdots,M_b\} \backslash j} m_{X_i^\tau \to g_i}(x_i^\tau), \\ &\propto \int_{\alpha_l} \sum_{\mathbf{x}_i^{M_b} \backslash x_i^j} g_i(y_i; Q[x_i], \alpha_l) \frac{m_{A_l \to g_i}(\alpha_l)}{p(\alpha_l|\mathbf{y}_l)} p(\alpha_l|\mathbf{y}_l) \prod_{\tau \in \{1,2,\cdots,M_b\} \backslash j} m_{X_i^\tau \to g_i}(x_i^\tau), \\ &\propto \mathrm{E}_{\alpha_l p(\alpha_l|\mathbf{y}_l)} \left[\sum_{\mathbf{x}_i^{M_b} \backslash x_i^j} g_i(y_i; Q[x_i], \alpha_l) \frac{m_{A_l \to f_i}(\alpha_l)}{p(\alpha_l|\mathbf{y}_l)} \prod_{\tau \in \{1,2,\cdots,M_b\} \backslash j} m_{X_i^\tau \to g_i}(x_i^\tau) \right]. \end{aligned} \tag{9}$$

where $\sum_{\mathbf{x}_i^{M_b} \backslash x_i^j}$ represents a sum over all the bit variables in $\mathbf{x}_i^{M_b}$ except x_i^j.

Then, the above message can be approximated by a list of N_p number of particles as

$$\tilde{m}_{g_i \to X_i^j}(x_i^j) \propto \frac{1}{N_p} \sum_{k=1}^{N_p} \sum_{\mathbf{x}_i^{M_b} \backslash x_i^j} g_i(y_i; Q[x_i], \alpha_l^k) \frac{\tilde{m}_{A_l \to g_i}(\alpha_l^k)}{p(\alpha_l^k|\mathbf{y}_l)} \prod_{\tau \in \{1,2,\cdots,M_b\} \backslash j} m_{X_i^\tau \to g_i}(x_i^\tau), \tag{10}$$

where $g_i(y_i; Q[x_i], \alpha_l^k)$ is shown in (3), $p(\alpha_l^k|\mathbf{y}_l)$ corresponds to the belief of each label $b(\alpha_l^k)$ and $b(\alpha_l^k) \propto \tilde{m}_{h_l \to A_l}(\alpha_l^k) \prod_{i \in \mathcal{N} \backslash h_l(A_l)} \tilde{m}_{g_i \to A_l}(\alpha_l^k)$. The candidate decoded $Q[x_i]$ from LDPC-based joint bit-plane SW decoding provided for correlation parameter estimation that corresponds to the message $m_{g_i \to A_l}(\alpha_l^k)$ with $k = 1, 2, \cdots, N_p$ can be written as

$$\tilde{m}_{g_i \to A_l}(\alpha_l^k) \propto \sum_{\mathbf{x}_i^{M_b}} g_i\left(\alpha_l^k, Q[x_i], y_i\right) \prod_{j \in \{1,2,\cdots,M_b\}} m_{X_i^j \to g_i}\left(x_i^j\right). \tag{11}$$

Thus, by performing message passing iteratively on the built factor graph, joint bit-plane decoding and accurate correlation estimation can be performed simultaneously to achieve efficient compression and good performance.

The message passing schedule for the whole factor graph in Fig. 1 is summarized in the following:

1) Draw the initial N_p number of variable particles in Region I from the priori distribution.
2) Initialize messages sent from factor nodes g_i to variable nodes X_i^j, $j = 1, 2, \cdots, M_b$, in Region III as in (10), for each of the M_b bit-planes.
3) If the decoded estimate (6) has the same syndrome as the received one or the maximum number of iterations is reached, export the decoded codeword and finish. If not, go to Step 4.
4) Update variable nodes in Region III according to the standard BP update rule given by (4) for SW decoding.
5) Compute the belief for each variable in Region III being $x_i^j \in \{0, 1\}$, $i = 1, 2, \cdots, N$, $j = 1, 2, \cdots, M_b$ according to (6).

6) Update particles in Region I by computing the belief (=weight) of each particle for each variable node in Region I. Here the rule for computing the belief of each particle is similar to (6) just replacing x_i to α_l^k.
7) Systematic resampling of particles in Region I, followed by the Metropolis-Hastings algorithm [29] and resetting the weight of particles to a uniform weight.
8) Update factor nodes in Regions I, II and III according to the rule in (5). Here, the specific update equations for factor nodes g_i in Region II are shown in (10) and (11).
9) Generate a new estimate based on the belief of variable nodes in Region III.
10) Go back to Step 3.

V. Transform-domain based DVC

In order to demonstrate the benefit of the proposed correlation tracking concept, we apply adaptive correlation estimation on the SW factor graph in a transform-domain state-of-the-art DVC codec. As in [6], [11], [30], all frames are classified into key frames or WZ frames, all even frames being WZ frames. The key frames are conventionally intra-coded, for example, using H.264 Advanced Video Coding (AVC) [31] intra coding mode. WZ frames are first transformed through a 4×4 discrete cosine transform (DCT) and the resulting transformed coefficients of WZ frame are grouped into 16 coefficient bands $b \in [1, 16]$. Then each coefficient band is quantized separately with 2^{M_b} quantization levels following [14]. Combination of all the bits from the quantization coefficient of each band is compressed using LDPC codes. For the WZ frame decoding, the transformed coefficients of each band will be decoded in our proposed joint bit-plane WZ decoder with PBP correlation estimation and side information available at the decoder which is described in Section IV.

In DVC, the side information frame, as the initial estimate of the current WZ frame generated based on the temporally adjacent reference frames, significantly influences the performance of WZ frame decoding. In our DVC setup, we generate side information frames according to [14], [32], [30], which perform motion-compensated interpolation (MCI) on the two closest reference frames of the current WZ frame via forward motion estimation, bidirectional motion estimation, spatial motion smoothing and bidirectional motion compensation.

In addition, letting the decoder control bitrate through a rate-adaptive scheme with a feedback channel is another way to improve the final compression performance in the state-of-the-art DVC codec, where some attractive solutions are given in [33], [34]. In this paper, we perform Low-Density Parity-Check Accumulate (LDPCA) encoding and embed rate-compatible LDPCA codes with a feedback channel in our proposed scheme for rate adaptation.

With the above design, we can demonstrate the benefits of our proposed scheme within a state-of-the-art DVC system and compare it with reference DVC systems from the literature such as DISCOVER [14]. Note that, in the proposed scheme, DCT coefficients, which need to be connected to the same correlation variable, may not be those neighboring variables like in the pixel-domain realization [25]. Thus it is necessary to build an adaptive connection between different DCT coefficients and correlation variables according to an initial correlation estimate obtained from the online estimate methods [11]. In the proposed scheme, DCT coefficients that have similar online estimated correlation statistics will be grouped together. In fact, we sort the online estimate obtained by using [11] first, and then every C successive estimates will be connected together. In this case, even though the connection ratio C is a constant, the connected variables are dynamically determined for different frames. Note that, the proposed scheme can easily be combined with any other correlation estimation techniques [12], [13], that would be treated as an initial estimate, to refine the estimated correlation.

Remark 1. The complexity of standard LDPC decoding can be reduced by passing log-likelihood ratios $L_{ai} \triangleq \log \frac{m_{a\to i}(0)}{m_{a\to i}(1)}$ instead of probabilities as messages. Note that the same method cannot be used in general for factor nodes in Regions I and II since the method can only be used for variables with alphabet size of two and there are generally more than two labels for the variables there. Indeed, the most complex part in the proposed adaptive decoding scheme is the use of the particle method and factor node update in Region II. However, the particle method is only used in Region I, and in fact, we do not need to perform the PBP algorithm in Region I at every BP iteration. Instead, we can start the PBP algorithm after some number of BP iterations and then perform a PBP iteration after every fixed number of BP iterations. Moreover, the particle method used in Region I updates each variable node locally, which means that the proposed PBP algorithm can easily take advantage of parallelism offered by Graphic Processing Unit (GPU) devices to speed up decoding (see [35], for example).

VI. Experimental Results

To verify the performance of correlation tracking across WZ-encoded frames in a video sequence, we tested the above setup with many standard QCIF 15Hz video sequences, “Soccer”, “Coastguard”, “Foreman”, “Hall & monitor”and “Salesman”. These videos covered fast, median and slow motion conditions. In the video sequences of our experiments, we consider two frames per group of picture (GOP) and a total of 149 frames for each video sequence. The odd frames are the key frames which are

TABLE I

AVERAGE RESULTS IN TERMS OF BJØNTEGAARD DELTA PSNR AND BITRATE METRIC FOR THE FIVE TEST SEQUENCES AND THREE DIFFERENT QUANTIZATION MATRICES.

		BJM	
		Δ PSNR (dB)	Δ Rate (%)
Foreman	Q1	-0.1341	-6.0592
	Q3	-0.2327	-10.5428
	Q5	-0.2576	-14.1864
Soccer	Q1	-0.4253	-16.1156
	Q3	-0.0104	-14.6673
	Q5	-0.0315	-18.1785
Coastguard	Q1	-0.0384	-0.0271
	Q3	-0.7409	-11.7729
	Q5	-0.9804	-18.9212
Hall	Q1	-0.2010	-6.1717
	Q3	-0.2609	-7.5663
	Q5	-0.8541	-12.8799
Salesman	Q1	-0.0318	-5.1683
	Q3	-0.0971	-7.9557
	Q5	-0.3693	-10.5094

intraframe coded and reconstructed using H.264/AVC with profile used in [14]. The even frames, between two intra coded key frames, are WZ frames which are encoded using LDPCA codes and recovered through our proposed joint bit-plane decoding with correlation estimation described in Section IV.

The test conditions for the WZ frames are described in the following. First, a 4×4 float DCT transform is performed, and the uniform scalar quantizer with data range $[0, 2^{11})$ and a dead-zone quantizer with doubled zero interval and the dynamic data range $[-\text{MaxVal}_b, \text{MaxVal}_b)$ are applied for DC and AC bands [14], respectively. At the decoder, the side information frame is generated using the recovered frames from H.264/AVC decoding, where search range with ± 32 pixels is used for the forward motion estimation [14]. The following parameters are used in our simulation: the number of particles $N_p = 12$, connection ratio $C = 4$ and random walk step $\sigma_r = 0.005$. In the proposed PBP algorithm, we do not assume that the initial LDPCA code rate is sufficiently large for successful decoding. In practice, it is difficult to allocate a suitable LDPCA code rate before decoding, since the true correlation information is unavailable. For the purpose of rate saving, the LDPCA decoder in feedback-based DVC schemes usually tries to decode

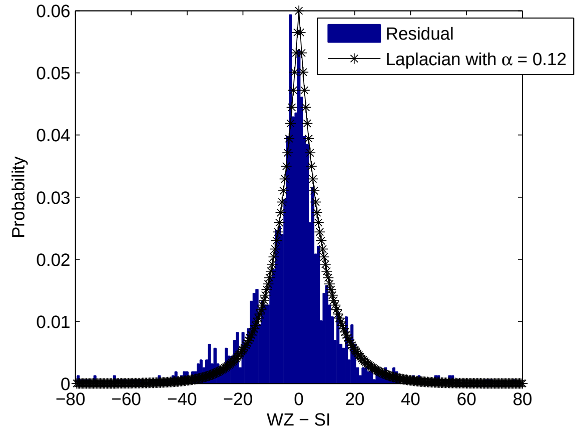

Fig. 2. Residual histogram for Foreman sequence at 15 Hz (DCT domain - DC coefficient band)

sources at a low code rate first, and then gradually increases code rate until successful decoding achieved. In the proposed PBP algorithm, if the LDPCA decoder cannot decode the source within a fixed number of iterations at a low code rate, it will opt for the PBP algorithm for online correlation estimation first, instead of directly increasing the code rate. In our simulations, the PBP algorithm is used only after 50 BP iterations and then is performed every 20 BP iterations.

In Fig. 2, we verified the Laplacian assumption of the correlation between the WZ frame and the side information frame. In this example, by setting $\alpha = 0.12$, Laplace distribution provides an accurate approximation to the residual between the WZ frame and side information frame.

First, we use the Bjøntegaard delta metric (BJM) [36] to illustrate the average difference between two rate-distortion curves in terms of PSNR or bitrate. The Bjøntegaard delta measurements of our proposed joint bit-plane decoding PBP method relative to the DISCOVER decoding with online estimation of [11] (benchmark) are given in Table I, where the simulations are performed on all test sequences and for three different quantization matrices Q1, Q3 and Q5 used in the DISCOVER codec [14]. The results show that our proposed algorithm always outperforms the benchmark decoder in terms of bitrate saving.

Results comparing the relative performance of DISCOVER (with the correlation estimator of [11]), joint bit-plane PBP (our proposed transform-based codec with adaptive correlation estimation), joint bit-plane off-line (joint bit-plane decoding using true correlation statistics estimated off-line [11]) and joint bit-plane on-line (joint bit-plane decoding using correlation statistics estimated on-line [11]) codecs for the Soccer, Coastguard, Foreman, Hall & Monitor and Salesman video sequences, respectively, are shown in Figs. 3 to 7. Note that the off-line estimation method of [11] models the correlation noise as a Laplacian distributed variable whose true Laplacian parameter is calculated off-line at the DCT-band/coefficient level

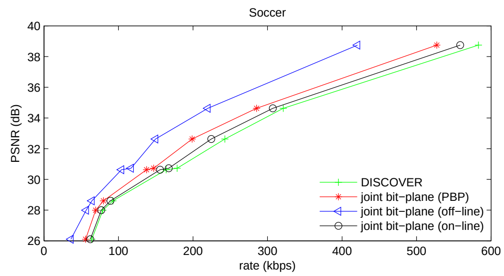

Fig. 3. PSNR comparison of the proposed PBP joint bit-plane DVC for the QCIF Soccer sequence, compressed at 15 fps.

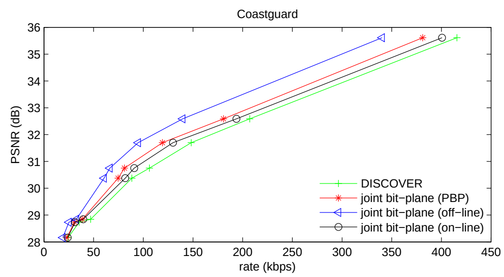

Fig. 4. PSNR comparison of the proposed PBP joint bit-plane DVC for the QCIF Coastguard sequence, compressed at 15 fps.

for each frame using the residual between the WZ frame and the side information. This is impractical since in this case the encoder would need to perform side information generation. On the other hand, the on-line estimation method of [11] models the correlation noise as Laplacian distributed, when the Laplacian parameter is estimated using the difference between backward and forward motion compensated frames at the decoder.

As expected, the joint bit-plane decoder (off-line) always achieves the best performance for all sequences (slow, median and fast motion), since it knows the true correlation statistics between source and side information. Most importantly, our proposed PBP-based codec consistently has the next best performance for all sequences, clearly outperforming the benchmark DISCOVER codec for all sequences since our PBP estimator iteratively refines the correlation statistics. We also note that the joint bit-plane setup shows a better performance than that of the benchmark DISCOVER codec since each code bit may obtain more information from its neighboring bit-planes than the traditional separate bit-plane decoding setup[2].

[2]Note that the LDPC code length in the joint bit-plane decoder is M_b times larger than that of separate bit-plane decoder, where M_b is the quantization level of DCT band b. Since the decoding performance of LDPC code depends on the code length, the joint bit-plane decoder with longer code length is expected to outperform a separate bit-plane decoder in the simulation. Another reason for this behavior is that the factor g_i in joint bit-plane decoder could obtain information from M_b variable nodes in Region III simultaneously, while such an advantage is not available in the separate bit-plane decoder.

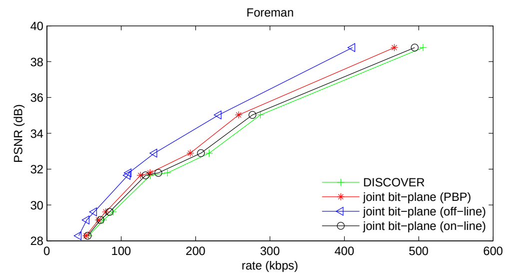

Fig. 5. PSNR comparison of the proposed PBP joint bit-plane DVC for the QCIF Foreman sequence, compressed at 15 fps.

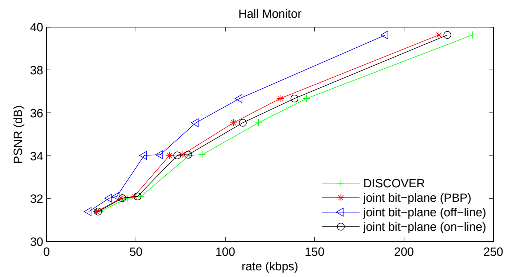

Fig. 6. PSNR comparison of the proposed PBP joint bit-plane DVC for the QCIF Hall Monitor sequence, compressed at 15 fps.

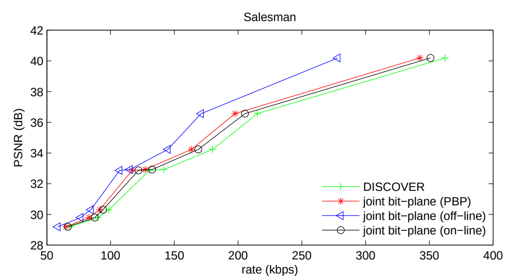

Fig. 7. PSNR comparison of the proposed PBP joint bit-plane DVC for the QCIF Salesman sequence, compressed at 15 fps.

A frame-by-frame PSNR variation for the Soccer sequence with quantization matrix Q8 is shown in Fig. 8. We found that the PSNR variation across frames is roughly 4.48 dB for the proposed PBP codec and 2.81 dB for H.264/AVC Intra codec. Although, the PSNR variation of the proposed codec is for almost 1.7dB larger than that of H.264/AVC Intra, the difference of average PSNR between H.264/AVC Intra and PBP codec is only 0.35 dB. Note that the Soccer sequence is a high-motion sequence, not suitable for DVC coding, hence the DISCOVER codec lags behind H.264 Intra for almost 2dB at this compression rate (see [14] and references therein). Fig. 8 shows that the PSNR fluctuations of PBP and H.264/AVC Intra have similar trend and the maximum PSNR difference between PBP and H.264 Intra curves is about 1.33 dB.

The estimation accuracy is studied in Fig. 9 for the DC band of the first WZ frame of the Soccer sequence. Assuming that the offline estimation is the benchmark for ideal correlation estimation, we

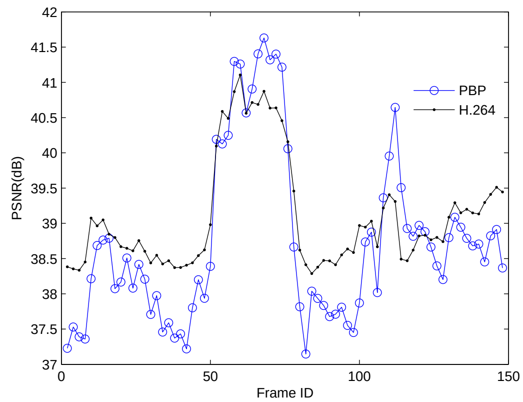

Fig. 8. Frame-by-frame PSNR variance for Soccer sequence with quantization matrix Q8.

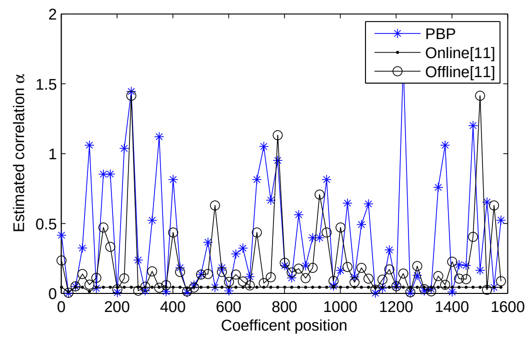

Fig. 9. Estimation accuracy of the proposed PBP method for the DC band of the first WZ frame of the Soccer sequence.

observe the online estimation [11] is consistently far off from the offline estimation [11] while the proposed PBP algorithm shows a similar tracking trend to the latter in estimating correlation. Similar results are obtained for other frames. This indicates the robustness of the correlation estimator even for high motion sequences such as Soccer.

Finally, the decoding algorithm complexity in terms of the execution time for both joint bit-plane decoding with online correlation estimation of [11] and the proposed joint bit-plane PBP algorithm is compared to that of the benchmark DISCOVER codec in Table II for two test sequences, WZ frames only, and three different quantization matrices. Encoding time is the same for all three methods. The simulations were conducted on a PC Intel i-7 980 CPU with 4GB memory. Joint bit-plane decoding with online correlation estimation of [11] shows comparable execution time with the benchmark DISCOVER codec. Although our current proof-of-concept proposed PBP decoding (implemented by MATLAB incorporating JAVA) needs longer execution time to achieve a better decoding performance in terms of bitrate saving, a shorter execution time is expected in future implementations by using more efficient programming (e.g.,

TABLE II

DECODING EXECUTION TIME (FULL SEQUENCE IN SECONDS) OF JOINT BIT-PLANE ON-LINE [11], JOINT BIT-PLANE PBP CODEC AND DISCOVER CODEC.

Sequences	DISCOVER	Joint bit-plane on-line	Joint bit-plane PBP
Foreman Q1	347.9376	359.9462	1266.4
Q3	457.0563	481.8287	3159.1
Q5	508.5244	631.7438	6035.6
Soccer Q1	581.0807	586.4243	2527.3
Q3	615.0184	623.4325	4634.3
Q5	657.3840	699.3597	6847.5

C++) and exploiting parallel processing inherent in the augmented graph decoding algorithm (see Remark 1), which would significantly mitigate the effect of particle-based BP.

VII. CONCLUSION

This paper proposes an adaptive correlation estimation scheme for distributed video coding. Unlike current work in that direction, our proposed technique is embedded within the SW decoder itself, thus ensuring true dynamic tracking of correlation estimation taking into account the variance of side information. This is achieved by augmenting the SW code factor graph with correlation parameter variable nodes together with additional factor nodes that connect the SW graph with the correlation variable nodes. In our examples, correlation is modeled as Laplacian noise although we note that other correlation models including Gaussian may be used with minimal change to the factor graph. Inference on the graph with multiple connected regions can then be achieved with standard belief propagation (BP) together with particle filtering that allows correlation parameter to take real values. The correlation variable nodes incorporate particles on which the particle filter operates, but also require joint operation with the BP algorithm which updates the weights of the particles. The proposed scheme boasts accurate correlation estimation together with ease of integration with existing DVC codecs, as all that is required is replacing the SW decoder block.

We demonstrate the benefit of using the proposed scheme with a state-of-the-art transform-domain based DVC using adjacent H.264/AVC compressed frames to generate side-information through motion-compensated interpolation (MCI). Simulation results for a range of slow to fast motion sequences show significant performance improvement due to correlation tracking by our proposed PBP algorithm over state-of-the-art DVC codec with correlation estimation. We also show that correlation statistics are accurately estimated online as the performance of our PBP algorithm closely (within 1-2dB PSNR) tracks that of the DVC codec which has offline knowledge of exact correlation statistics.

REFERENCES

[1] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. 19, pp. 471–480, Jul. 1973.

[2] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. 22, pp. 1–10, Jan. 1976.

[3] A. Wyner, "Recent results in the Shannon theory," *IEEE Trans. Inform. Theory*, vol. 20, pp. 2–10, Jan. 1974.

[4] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (discus): Design and construction," in *Proc. DCC-1999 Data Compression Conf.*, 1999, pp. 158–167.

[5] Z. Xiong, A. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Process. Magazine*, vol. 21, pp. 80–94, Sep. 2004.

[6] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," in *the Thirty-Sixth Asilomar Conf. Signals, Syst. Computers*, vol. 1, 2002, pp. 240–244.

[7] R. Puri and K. Ramchandran, "PRISM: A new robust video coding architecture based on distributed compression principles," in *Proc. Annual Allerton Conf. Commun., Control and Computing*, vol. 40, no. 1, 2002, pp. 586–595.

[8] Y. Yang, V. Stankovic, Z. Xiong, and W. Zhao, "On multiterminal source code design," *IEEE Trans. Inform. Theory*, vol. 54, no. 5, pp. 2278–2302, 2008.

[9] P. Meyer, R. Westerlaken, R. Gunnewiek, and R. Lagendijk, "Distributed source coding of video with non-stationary side-information," in *Proc. SPIE*, vol. 5960, 2005, pp. 857–866.

[10] M. Dalai, R. Leonardi, and F. Pereira, "Improving turbo codec integration in pixel-domain distributed video coding," in *Proc. ICASSP-2006 IEEE Int. Conf. Acoustics, Speech and Sig. Proc.*, vol. 2, pp. II–II.

[11] C. Brites and F. Pereira, "Correlation noise modeling for efficient pixel and transform domain Wyner–Ziv video coding," *IEEE Trans. Circuits and Sys. Video Techn.*, vol. 18, no. 9, pp. 1177–1190, 2008.

[12] X. Fan, O. Au, and N. Cheung, "Adaptive correlation estimation for general Wyner-Ziv video coding," in *Proc. ICIP-2009 IEEE Int. Conf. Image Proc.*, 2009, pp. 1409–1412.

[13] X. Huang and S. Forchhammer, "Improved virtual channel noise model for transform domain Wyner-Ziv video coding," in *Proc. ICASSP-2009 IEEE Int. Conf. Acoustics, Speech and Sig. Proc.*, 2009, pp. 921–924.

[14] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, "The DISCOVER codec: architecture, techniques and evaluation," in *PCS-2007 Picture Coding Symposium*, 2007.

[15] D. Varodayan, A. Mavlankar, M. Flierl, and B. Girod, "Distributed grayscale stereo image coding with unsupervised learning of disparity," in *Proc. DCC-2007 IEEE Data Compression Conf.*, 2007, pp. 143–152.

[16] Y. Fang, "Crossover probability estimation using mean-intrinsic-LLR of LDPC syndrome," *IEEE Communications Letters*, vol. 13, no. 9, p. 679, 2009.

[17] A. Zia, J. Reilly, and S. Shirani, "Distributed Parameter Estimation with Side Information: A Factor Graph Approach," in *Proc. ISIT-2007 IEEE Int. Symp. Inform. Theory*, 2007, pp. 2556–2560.

[18] V. Toto-Zarasoa, A. Roumy, and C. Guillemot, "Maximum Likelihood BSC parameter estimation for the Slepian-Wolf problem," *IEEE Communications Letters*, vol. 15, no. 2, 2011.

[19] S. Cheng, S. Wang, and L. Cui, “Adaptive slepian-wolf decoding using particle filtering based belief propagation,” in *Proc. the 47th Annual Allerton Conf.*, Oct 2009.

[20] L. Cui, S. Wang, S. Cheng, and Q. Wu, “Noise adaptive ldpc decoding using particle filters,” in *Proc. CISS 2009*, Baltimore, MD, 2009.

[21] B. Girod, A. M. Aaron, S. Rane, and D. A. R.-M. D. Rebollo-Monedero, “Distributed video coding,” *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, 2005.

[22] C. Guillemot, F. Pereira, L. Torres, T. Ebrahimi, R. Leonardi, and J. Ostermann, “Distributed monoview and multiview video coding: basics, problems and recent advances,” *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 67–76, 2007.

[23] L. Stankovic, V. Stankovic, and S. Cheng, “Distributed compression: Overview of current and emerging multimedia applications,” in *Proc. ICIP-2011 IEEE Int. Conf. Image Processing*, Sept. 2011.

[24] N. Deligiannis, A. Munteanu, T. Clerckx, P. Schelkens, and J. Cornelis, “Modeling the correlation noise in spatial domain distributed video coding,” in *Proc. DCC-2009 Data Compression Conf.*, 2009, p. 443.

[25] L. Stankovic, V. Stankovic, S. Wang, and S. Cheng, “Distributed video coding with particle filtering for correlation tracking,” in *Proc. Eusipco-2010 18th European Signal Proc. Conf.*, Aug. 2010.

[26] F. Kschischang, B. Frey, and H. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[27] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Trans. Signal Proc.*, vol. 50, no. 2, pp. 174–188, 2002.

[28] M. Bolic, P. M. Djuric, and S. Hong, “Resampling algorithms for particle filters: A computational complexity perspective,” *EURASIP J. Applied Signal Proc.*, vol. 15, no. 5, pp. 2267–2277, 2004.

[29] S. Chib and E. Greenberg, “Understanding the Metropolis-Hastings algorithm,” *American Statistician*, vol. 49, no. 4, pp. 327–335, 1995.

[30] J. Ascenso, C. Brites, and F. Pereira, “Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding,” in *5th EURASIP Conf. Speech, Image Proc., Multimedia Commun., Services*, 2005.

[31] I. T. U. Recommendation, “Advanced Video Coding for Generic Audiovisual Services,” *ITU-T Rec. H.264*, 2007.

[32] J. Ascenso, C. Brites, and F. Pereira, “Content adaptive Wyner-Ziv video coding driven by motion activity,” in *Proc. ICIP-2006 IEEE Int. Conf. Image Processing*, 2006, pp. 605–608.

[33] J. Chen, A. Khisti, D. Malioutov, and J. Yedidia, “Distributed source coding using serially-concatenated-accumulate codes,” in *Proc. ITW-2004 IEEE Inform. Theory Workshop*, 2004, pp. 209–214.

[34] D. Varodayan, A. Aaron, and B. Girod, “Rate-adaptive codes for distributed source coding,” *Signal Processing*, vol. 86, no. 11, pp. 3123–3130, 2006.

[35] S. Wang, L. Cui, S. Cheng, and R. Huck, “Noise adaptive LDPC decoding using particle filtering based on parallel gpu architecture,” *2009 NVIDIA Research Summit*, Oct. 2009.

[36] G. Bjontegard, “Calculation of average psnr differences between rd-curves,” *ITU-T VCEG-M33*, 2001.