Distributed Video Coding with Progressive Significance Map

Yang Hu[1], William A. Pearlman*

Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

Abstract

A distributed video coding (DVC) system based on wavelet transform and set partition coding (SPC) is presented in this paper. Conventionally the significance map (sig-map) of SPC is not conducive to Slepian-Wolf (SW) coding, because of the difficulty of generating a side information sig-map and the sensitivity to decoding errors. The proposed DVC system utilizes a higher structured significance map, named progressive significance map (prog-sig-map), which structures the significance information into two parts: a high-level summation significance map (sum-sig-map) and a low-level complementary significance map (comp-sig-map). This prog-sig-map alleviates the above difficulties and thus makes part of the prog-sig-map (specifically, the fixed-length-coded comp-sig-map) suitable for SW coding. Simulation results are provided showing the improved rate-distortion performance of the DVC system even with a simple system configuration.

Keywords: Distributed video coding, set partition coding, SPIHT coding,

*Corresponding author

Email addresses: yanghuyh@gmail.com (Yang Hu), pearlw@ecse.rpi.edu (William A. Pearlman)

[1]Y. Hu is now with Cisco Systems, Santa Clara, CA.

significance map, wavelet transform

1. Introduction

Distributed video coding (DVC) is a class of distributed source coding (DSC) applied to video source data. It is currently a subject of intense research. In a number of articles in the DSC literature, set partition coding (SPC), usually combined with wavelet transform, is used as the source coder preparing the data for Slepian-Wolf (SW) coding. SPC is a widely used technique for image and video compression, used in the JPEG 2000 [1, 2] image compression standard and the new 2D lossy-to-lossless compression algorithm recently standardized by the Consultative Committee for Space Data Systems (CCSDS) [3, 4]. It has also been successfully applied to several other well-known image coding algorithms including SPIHT [5], SPECK [6], SWEET [7], SBHP [8], and EZBC [9]. A comprehensive tutorial on SPC and its usage in wavelet coding systems can be found in the monographs [10, 11] and the textbook [12].

A salient feature of SPC is that its compressed bitstream divides naturally into two components: a significance map (sig-map) that conveys location information; and a value bitstream that conveys intensity information of signs and lower order bits of wavelet coefficients. According to the DVC tradition, the module that reflects the DSC principle is referred to as a Slepian-Wolf codec, which includes an SW encoder and an SW decoder. The frame to be SW coded is referred to as an SW frame. In most cases of DSC, only the value data of an SPC bitstream are SW coded. For example, Guo et.al. [13] codes the SW frame with single-pass ZTE [14] coding, where the significance map is coded in normal intra-frame or intra mode (sent out directly without SW coding), and only significant coefficients are further SW coded. A similar strategy is used when the SW frames are SPIHT coded [15, 16]. In another work, a modified quadtree scanning is applied, wherein the decision bits from the quadtree scanning, similar to the significance map, is still intra coded [17].

The significance map occupies about 60% of the SPC bitstream for bit rates giving recovery quality measured by PSNR's in the 30 to 40 dB range, which covers almost all video viewing experiences. An efficient DSC system should try to SW code the sig-map in addition to the value information. However, the conventional sig-map is not conducive to SW coding, mainly because of two difficulties.

- It is hard or impossible to generate a qualified side information (SI) sig-map at the decoder, which should be of the same length as the original sig-map and be simultaneously bit-wise correlated with the original one.

- The conventional sig-map is highly error sensitive. Decoding errors are unavoidable for SW decoder even though the error rate could be very low. Due to error propagation and the consequent loss of coding synchronization, even a single decoding error can be catastrophic.

To alleviate these difficulties, there are two existing strategies, both of which try to express (part of) the significance locations with fixed-length codes. In the first strategy the SPC is abandoned, and the significance locations, which are the locations of significant sets and pixels, are coded via a position-by-position scan of the DWT coefficients [18]. The second strategy extracts a fixed-length-coded part of the location information and SW codes just that part. For example, one method [19] separates the sig-map of SPIHT into two parts called SD and SP. The SD information, indicating whether or not a tree is a zerotree, is intra coded (without SW coding). For non-zerotrees, SP bits are sent out indicating the significance of every tested subset (coefficients and/or descendant subtrees) in the current tested tree. In this way, the SP is of fixed length. The decoder can produce side information SP′ with the help of SD. With either strategy, the sig-map alleviates the difficulty of SI generation, thanks to the fixed length property. Such a sig-map however still faces the hurdle of loss of synchronization. Even a single bit error in the decoded sig-map (or the SP) may destroy the whole decoding that follows.

The progressive significance map (prog-sig-map) [20] introduces a layered structure to represent the significance information. It consists of a summation significance map (sum-sig-map) and a *fixed-length-coded* complementary significance map (comp-sig-map. The sum-sig-map occupies about 60% and the comp-sig-map about 40% of the prog-sig-map for the common video bit rates. Thanks to the constraint of the sum-sig-map and the fixed-length property of the comp-sig-map, the decoding errors associated with the comp-sig-map would not propagate within the comp-sig-map or to the decoding of the sum-sig-map. In this paper, we propose a wavelet-based DVC system using SPC with the prog-sig-map to make both the comp-sig-map (the fixed-length part of the prog-sig-map) and the value information suitable for SW coding.

To our knowledge, there was only one previous attempt to SW encode the significance map. In this work [21], the significance map was divided into bits from direct and indirect descendants, which were separately SW encoded in each bit plane. Sychronization of these relatively short bitstream segments can be maintained between the key and SW frames only when they are very highly correlated. For the multi-spectral images in their simulations, that was the case, but for video sequences, the target source in this paper, such synchrony could not be maintained. Thus, this method is inapplicable to video.

The rest of the paper is organized as follows. A brief overview of the progressive significance map is provided in Section 2. We introduce the proposed DVC system in Section 3. Modules in the system are detailed in Section 4. Section 5 provides the simulation results. Finally, Section 6 concludes the paper.

2. Overview the Progressive Significance Map

A significance map conveys the location information of significant sets and pixels, generated from the significance tests (sig-tests), and indicates the execution path of an SPC process. There are various set structures and partitioning rules in different SPC coders (such as the spatial orientation tree in SPIHT [5], the quadtree and octave banding partitioning in SPECK [6] and SWEET [7], etc.). They all use the core sig-test that can be uniformly referred to as the (c, w) test. The c denotes the number of *candidates*, defined as the sets (including singleton sets) to be sig-tested; the w denotes the number of *winners*, defined as the newly found significant sets in the sig-test. Conventionally each candidate is assigned a binary indicator I signalling whether or not it is a winner ($I = 1$, if winner; $I = 0$, otherwise). In this way, a (c, w) test generates c indicators $\{I_1, I_2, ..., I_c\}$, which go to the conventional sig-map.

The progressive sig-map also consists of the results of a series of correlated (c, w) tests, but in a layered structure, using a sum-sig-map and a comp-sig-map. The sum-sig-map conveys how many of the c candidates are winners. The number of winners w is actually the summation of the c indicators as $w = \sum_{i=1}^{c} I_i$. If $w = 0$ or $w = c$, the c indicators must be all 0's or all 1's, respectively. If $0 < w < c$, there are $M = \binom{c}{w}$ different choices for the w winners out of the c candidates. In this case, the comp-sig-map is introduced. For most 2D SPC methods, the candidate number c satisfies $c \in \{1, 2, 3, 4\}$, so six types of (c,w) tests ($(2, 1)$, $(3, 1)$, $(3, 2)$, $(4, 1)$, $(4, 2)$, and $(4, 3)$ tests) need a comp-sig-map. In the variable-length comp-sig-map, Huffman codes are used to code a set of M equally probable symbols for each (c, w) type. To build the fixed-length-coded comp-sig-map, additional bits are appended to the shorter Huffman codewords so as to make them of the same length as the longer ones for each (c, w) type. In this way, a (c, w) test has a fixed number $L(c, w)$ of complementary bits, according to $L(c, w) = \lceil \log_2 M \rceil$.

The fixed-length feature of the comp-sig-map enables the generation of a qualified SI comp-sig-map. In addition, if the sum-sig-map is correct, it will produce correct winner numbers. Under the condition of correct winner numbers, the fixed-length comp-sig-map is robust to bit errors, in the sense that bit errors in the comp-sig-map will impact only locally instead of propagating within the comp-sig-map or to the sum-sig-map and value data. In this way, the prog-sig-map alleviates the drawbacks (SI generation difficulty and high sensitivity to decoding errors) of the conventional sig-map in DVC application. Thus, the fixed-length comp-sig-map is a good source for SW coding. For details concerning the compression performance of the prog-sig-map and the error resilience of the fixed-length comp-sig-map, please refer to its original paper [20].

3. Proposed DVC System

The proposed DVC system is illustrated in Figure 1. This system is an implementation of Wyner's DSC scheme[22] using punctured or rate adaptive low-density parity check (LDPC) codes. Explanations of Wyner's scheme and principles of distributed source coding may be found in the textbook by Pearlman and Said[12]. Different from existing DSC schemes using coders that generate significance maps (sig-maps), part of the sig-map, the fixed-length part called the comp-sig-map, serves as the SW coding object and is robust to decoding errors. In order to highlight the key point (the SW coding of the comp-sig-map), we adopt very simple techniques for other modules. A video sequence is split into *key frames* and *SW frames*. The key frames are coded with conventional intra coding methods, such as H.264 intra coding, JPEG 2000, normal 2D SPIHT, and so forth. The SW frames are coded in a distributed way. A GOF (group of frames) of length 2 is used — the even frames are key frames and the odd frames are SW frames.

An SW frame is first set partition coded to generate its prog-sig-map, comprising the sum-sig-map S and the fixed-length comp-sig-map M, and significant value information V, comprising signs and refinement bits. The sum-sig-map S is sent to the output bitstream directly; the fixed-length comp-sig-map M and the value information V are SW coded, *separately*.

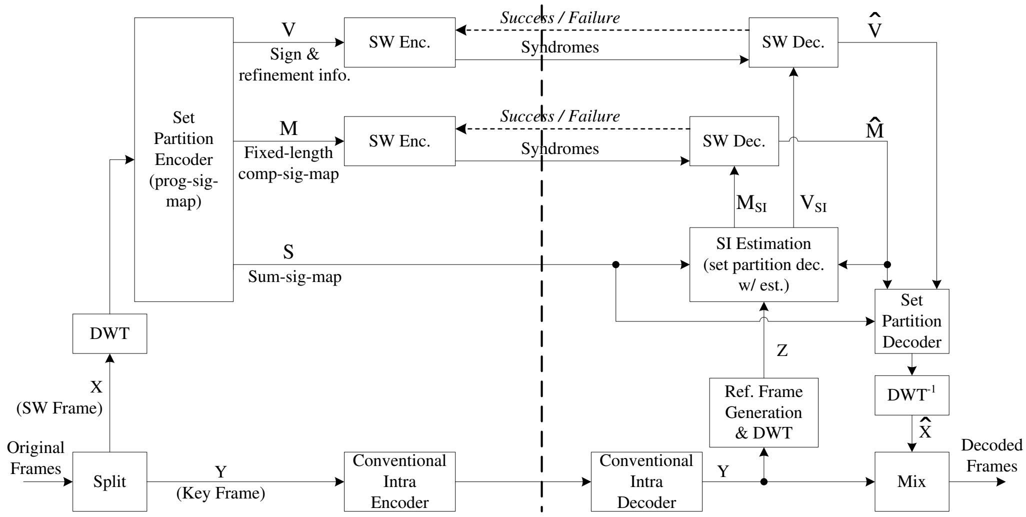


Figure 1: Block diagram of the proposed DVC system based on SPC with prog-sig-map.

An SW encoder includes a channel encoder and a parity generator. The channel encoder computes a set of syndrome bits. These syndrome bits are buffered and incrementally sent to the decoder which iteratively asks for augmented syndrome bits via the feedback channel until successful decoding is achieved. The parity generators calculate error-checking bits, that are sent to the decoder to help detect false positive decoding.

The decoding of a SW frame is more complex. The key frames are assumed to be losslessly available at the decoder. An estimation of the original SW frame is calculated from the available key frames. This estimated frame is called a *reference frame*, which is exploited when generating the side information for both the comp-sig-map and the value data. Although there are many sophisticated estimation methods to produce the reference frame, for simplicity, we average the co-located pixels of the key frames right before and after the current SW frame. The frame consisting of these average values serves as the reference frame Z in our system.

The significance map of the SW frame is first decoded. Within the significance map, the sum-sig-map S is received directly and decoded conventionally. The SW-coded comp-sig-map M needs to be SW-decoded. Recall that M comprises the syndrome of the comp-sig-map, so conveys only signals in a specific coset of the channel code. One must use the side-information M_{SI} to determine the most likely signal in the coset. Its side information M_{SI} is produced in the “SI Estimation” module, where M_{SI} is estimated via a *virtual* set partition decoding of the Z reference frame under the guidance of the sum-sig-map S. Specifically, for each significance test of the virtual decoding process, only candidate number c (inferred from previous tests) and winner number w (decoded from the sum-sig-map S) are available; the indicators of the sig-test must be estimated based on the co-located coefficients of the reference frame Z. Based on (c, w) and the estimated indicators, we get the corresponding comp-bits, which form the side information M_{SI} of the comp-sig-map. The details of this estimation are provided in Section 4.2. In the SW decoder, channel decoding (syndrome-adjusted LDPC decoding) is applied on M_{SI} adjusted by the received syndrome to recover the comp-sig-map $\hat{M}$. After each channel decoding and parity checking, the success or failure is signaled to the encoder, asking for the syndrome of the next codeword or asking for augmented syndrome bits until successful decoding is achieved. The recovered comp-sig-map $\hat{M}$ is fed back to the SI estimator, because the next M_{SI} codeword will be estimated upon the successfully decoded codewords.

After sig-map recovery, the decoder reconstructs the value data. Before the SW decoding of value data, the side information V_{SI} must be produced first. The SI estimator applies the decoded sum-sig-map S and $\hat{M}$ to the reference frame Z and extracts the signs and refinement bits from Z to form the side information V_{SI}. Then SW decoding is performed. Similar to the SW decoding of the comp-sig-map, channel decoding (syndrome-adjusted LDPC decoding) is applied on V_{SI} adjusted by the received syndrome to reconstruct the value information $\hat{V}$. After each channel decoding and parity checking, the success or failure is signaled to the encoder. Upon a decoding failure, augmented syndrome bits will be transmitted to the decoder until successful decoding is achieved. Then the decoder continues decoding the next value codeword.

Finally, the normal set partition decoding is performed to recover the SW frame $\hat{X}$, using the sum-sig-map S, the comp-sig-map $\hat{M}$, and the value data $\hat{V}$.

4. System Modules

In this section, we describe the details of the important modules in the proposed DVC system.

4.1. SW Encoding of Comp-sig-map

The SW encoder includes a channel encoder and a parity generator. It embodies the distributed coding strategy, that is, the rate of the output data is lower bounded by $H(M|M_{SI})$ [23] instead of $H(M)$. The rate-adaptive LDPC accumulated (LDPCA) codes for distributed source coding [24] are used as the channel codes. The comp-sig-map M is blocked into n-bit words that are inputs to the SW encoder. For each input word, n syndrome bits are accumulated and buffered at the encoder and are sent incrementally.

The initial transmitted syndrome length should be decided by the SW boundary $H(M|M_{SI})$. Since the statistics between the source and the side information are not available at the encoder, the SW boundary must be estimated based on properly designed correlation model for M and M_{SI}. There are some existing techniques, such as the ones used in a state-of-the-art DVC codec DISCOVER [25]. Instead, we simply transmit from the mimimum number of syndrome bits ($n \times {}^{2}/_{66}$) supported by the LDPCA program code [26]. The program code provides a compression fraction ${}^{m}/_{n} \in \{{}^{2}/_{66}, {}^{3}/_{66}, ..., {}^{66}/_{66}\}$, where m is the number of transmitted syndrome bits and n is the input word length. Upon channel decoding failure, more syndrome bits (corresponding to the minimum increment ${}^{1}/_{66}$ of the compression fraction in LDPCA program code) are requested and transmitted, until successful decoding is obtained. When computing the compressed bit rate in our simulations, the feedback bits are not counted. To detect false positive LDPC decoding, six parity bits are computed and sent out for each codeword. They are the parities of the complementary bits from the six types of sig-tests ($(2,1)$, $(3,1)$, $(3,2)$, $(4,1)$, $(4,2)$, and $(4,3)$ tests), respectively.

4.2. SI Comp-sig-map Generation

The side information comp-sig-map M_{SI} is an estimation of the original comp-sig-map M. The estimation is directed by the sum-sig-map S (or equivalently w) and exploits the coefficient values of the reference frame Z. Specifically, a virtual decoding is performed under the indication of the sum-sig-map. In the virtual process, a sig-test can infer the candidate number c

from previous tests and get the winner number w from the sum-sig-map. A real decoding must know the indicators for the sig-test in order to properly perform set partitioning or value recovering; while the virtual one only knows that, among the c tested sets, there are w winners/significant sets. The virtual decoder must "guess" the locations of the w winners out of the c candidates, from the $\binom{c}{w}$ possible choices, and then generates the estimated c indicators accordingly. Based on (c,w) and the estimated indicators, we get the corresponding comp-bits, which form the side information M_{SI} of the comp-sig-map. This guess/decision is based on the co-located coefficients of the reference frame Z and is used to create the side information V_{SI} for SW-decoding the value data of the current SW frame.

4.2.1. A Simple Decision Strategy

We recall that the reference frame Z is an estimation of the current SW frame. The Z can be expected to be highly correlated/similar to the current SW frame X. Consider the following situation.

For frame X, there are c candidate coefficients or single element sets, with coordinates $\{S_1, S_2, ..., S_c\}$, of which w are significant. We want to estimate/decide which w of the c candidates are winners. Denote the co-located frame-Z coefficients as $\{Z(S_1), Z(S_2), ..., Z(S_c)\}$, among which $\{Z(S_{i_1}), Z(S_{i_2}), ..., Z(S_{i_w})\}$ have the maximum magnitudes. Since frame Z is quite similar to frame X, it is reasonable to decide that the frame-X coefficients located at $\{S_{i_1}, S_{i_2}, ..., S_{i_w}\}$ are the w winners out of the c candidates.

4.2.2. The Decision Strategy

We now present a more comprehensive decision strategy for choosing the locations of the winners. For a sig-test with c condidates, located at $\{S_1, ..., S_c\}$, and w winners, there are $\binom{c}{w}$ different choices for the w winners out of the c candidates. For each possible choice $m, m \in \{1, 2, ..., \binom{c}{w}\}$, the winners under the choice m are denoted by $\{S_{i_1(m)}, S_{i_2(m)}, ..., S_{i_w(m)}\}$, where $i_l(m)$, $l \in \{1, 2, ..., w\}$, indexes the lth possible winner under choice m.

When the candidates are single-element sets, the co-located frame-Z coefficients are $\{Z(S_{i_1(m)}), Z(S_{i_2(m)}), ..., Z(S_{i_w(m)})\}$. Define *average magnitude* $\bar{A}_m$ for choice m as

$$\bar{A}_m = \frac{\sum_{l=1}^{w} |Z(S_{i_l(m)})|}{w}. \tag{1}$$

The optimal choice $\hat{m}$ is the one with maximum $\bar{A}$ as

$$\hat{m} = \arg\max_m \{\bar{A}_m\}. \tag{2}$$

and the corresponding optimal average magnitude is $\hat{A} = \bar{A}_{\hat{m}}$.

Otherwise, when the candidates are multi-element sets, define the *average descendent magnitude* $\bar{B}_m$ of choice m as

$$\bar{B}_m = \frac{\sum_{l=1}^{w} \hat{B}(S_{i_l(m)})}{w}. \tag{3}$$

where $\hat{B}(S_{i_l(m)})$ is the average descendant magnitude of the optimal indicator choice for the subsets that are descended from $S_{i_l(m)}$. The optimal choice $\hat{m}$ is still the one having the maximum value as

$$\hat{m} = \arg\max_m \{\bar{B}_m\}. \tag{4}$$

and the corresponding optimal average descendant magnitude is $\hat{B} = \bar{B}_{\hat{m}}$. In summary, the decision for multi-element sets are based on the descendant magnitude $\bar{B}_m$ of each choice (as in (4)), while the descendant magnitude $\bar{B}_m$ is further based on the descendant magnitude of each possible winner $S_{i_l(m)}$, $l = 1, 2, ..., w$, (as in (3)).

4.2.3. Implementation

The estimation process of any set S (to be divided into single- or multi-element subsets) is implemented with the function **optimalB(S)**, defined in Algorithm 1. To obtain the optimal descendant magnitude $\hat{B}$ of a multi-element set $S_{i_l(m)}$, the function **optimalB(S)** is called recursively, as in the step $\hat{B}(S_{i_l(m)}) = \textbf{optimalB}(S_{i_l(m)})$. However, if $S_{i_l(m)}$ is a single-element set, it cannot be further set partitioned and consequently has no "real" descendant magnitude. In this case, we set $\hat{B}(S_{i_l(m)})$ as the magnitude of the co-located frame Z coefficient $|Z(S_{i_l(m)})|$, which agrees with the expression in (1). This case serves as the stop condition for the recursion.

Upon the decision of the optimal indicator choices for the sig-tests, the fixed-length comp-bits corresponding to the optimal choice $\hat{m}$ will go to M_{SI}. Because the sig-test order in this virtual decoding (producing M_{SI}) is consistent with that in the encoding process (producing original M), the estimated M_{SI} and the original M are bit-wise correlated and of the same length. If a frame X^* is recovered indicated by S and M_{SI}, it will be the one that is most similar to the reference frame Z among all possible recovered frames satisfying the sum-sig-map S.

Algorithm 1 optimalB(S)

- Set partition S into set group $\{S_1, ..., S_c\}$.
- Get w for the set group.
- For each possible choice $m \in \{1, 2, ..., \binom{c}{w}\}$, calculate the corresponding average descendant magnitude $\bar{B}_m$ as below.
  - For each possible winner $S_{i_l(m)}$, $l \in \{1, 2, ..., w\}$, under choice m, achieve its optimal descendant magnitude $\hat{B}(S_{i_l(m)})$ as below.
    * If $S_{i_l(m)}$ is a multi-element set,

$$\hat{B}(S_{i_l(m)}) = \mathbf{optimalB}(S_{i_l(m)})$$

    * Else, $S_{i_l(m)}$ is a single-element set,

$$\hat{B}(S_{i_l(m)}) = |Z(S_{i_l(m)})|$$

  - Calculate $\bar{B}_m = \frac{\sum_{l=1}^{w} \hat{B}(S_{i_l(m)})}{w}$.
- Decide $\hat{m} = \arg\max_m\{\bar{B}_m\}$ and set $\hat{B} = \bar{B}_{\hat{m}}$.
- Return $\hat{B}$.

4.3. SW Decoding of Comp-sig-map

The SW decoder includes an intrinsic Log-Likelihood Ratio (LLR) calculator, a channel decoder, and a decoding error detector. It aims at recovering the comp-sig-map $\hat{M}$ from the SI comp-sig-map M_{SI}.

The channel decoder performs LDPCA [24] decoding. The LDPC graph structure is properly modified each time it receives an additional increment of the syndrome. Then the syndrome-adjusted LDPC iterative decoding method [27] can be applied using the syndrome bits, the LDPC graph structure for the current syndrome, and the intrinsic LLRs of the source bits. The intrinsic LLR calculation is introduced later. The LDPC decoder is the most computationally intensive module in the whole proposed system.

Each LDPCA decoded codeword will be checked in the error detector. Upon successful channel decoding, three conditions should be satisfied.

- The LDPC syndrome.
- The parity bits.
- The (c, w) constraint and padding bit constraint.

An example of the (c, w) constraint is that, in a $(4, 2)$-test, the 3 fixed-length comp-bits could not be 000 or 111, which are not included in the codebook for the $(4, 2)$-test. In addition, when M or M_{SI} is blocked into codewords, the comp-bits from a test are not cut off to be assigned to two codewords, so some bits (such as 0's) are padded to the end of a codeword to make it of length n. The padding bits in the recovered codeword should be the same as those added originally.

After each LDPC decoding, the decoding failure or success is signaled to the encoder, asking for augmented syndrome bits or asking for the syndrome of the next codeword. In the worst case, all n syndrome bits are requested and transmitted to the decoder, in which case successful decoding is guaranteed [24].

4.3.1. Intrinsic LLR Calculation

To perform the syndrome-adjusted LDPC iterative decoding [27], the source nodes are seeded with the intrinsic LLR of the source bits. Then the messages are computed and passed back and forth between the source nodes and the syndrome nodes according to the equations in Section III-B of the original paper by Liveris et. al. [27] until the estimated source bits converge.

Let $\{x_1, x_2, ..., x_n\}$ denote the source bits, i.e., the code-bits in the original comp-sig-map M; let $\{y_1, y_2, ..., y_n\}$ denote the SI bits, i.e., the code-bits in the M_{SI} codeword. The intrinsic LLR of source bit i is defined as

$$\mathrm{LLR}_i = \log\frac{\Pr[x_i = 0|y_i]}{\Pr[x_i = 1|y_i]} = (1 - 2y_i)\log\frac{1 - p_i}{p_i} \tag{5}$$

where $p_i = \Pr[x_i \neq y_i|y_i]$ and $i = 1, 2, ..., n$. It is similar to Equation (1) of Liveris' paper, where the correlation channel between the source and SI bits is modeled as a binary symmetric channel (BSC), and thus $p_i = \Pr[x_i \neq y_i|y_i] = \Pr[x_i \neq y_i] = p$. However, in our application, the p_i is related to the test type (c, w). Define the following notations.

- $p_i^{(0)} = \Pr[x_i \neq y_i|y_i = 0]$
- $p_i^{(1)} = \Pr[x_i \neq y_i|y_i = 1]$
- (c_i, w_i) : the type of the sig-test generating bit i.

- p_e : the error probability of the (c_i, w_i) test.

For a (c, w) test, $p_i^{(0)}$ and $p_i^{(1)}$ can be expressed in terms of c, w, and p_e in the following equations.

$$p_i^{(0)} = \frac{\binom{c-1}{w-1}}{\binom{c}{w} - 1} \times p_e \tag{6}$$

$$p_i^{(1)} = \frac{\binom{c-1}{w}}{\binom{c}{w} - 1} \times p_e \tag{7}$$

The p_e for each test type can be estimated separately. For simplicity, we estimate p_e for all test types in the current codeword, by means of comparing the last K (such as 100) test units in M_{SI} with those in $\hat{M}$.

In summary, for each codeword, the test error probability p_e is first estimated. Given the M_{SI} code-bits $\{y_i\}$, $i = 1, 2, ..., n$, the intrinsic LLR for each source bit i is calculated according to (5), where

$$p_i = \begin{cases} p_i^{(0)}, & \text{if } y_i = 0; \\ p_i^{(1)}, & \text{if } y_i = 1. \end{cases} \tag{8}$$

and $p_i^{(0)}$ or $p_i^{(1)}$ is calculated from p_e according to (6) and (7), given the test type (c_i, w_i). Then the LLR_i, $i = 1, 2, ..., n$, are used by LDPC decoding (in the same way as the $q_{i,0}$ in Liveris' paper).

4.4. SW Bypass of Comp-sig-map

When decoding to the lower bit-planes, the correlation between original comp-sig-map M and its side information M_{SI} is quite small. This small correlation translates to a length of the SW syndrome output that approaches the length of the comp-sig-map input. Given the 3% to 5% penalty introduced by this fixed-length structure within the prog-sig-map, SW coding should be bypassed when it fails to achieve compression relative to arithmetic coding of the conventional sig-map. For this purpose, we define a compression fraction η to be the length of the syndrome output divided by the length of the comp-sig-map, which is expressed more formally as

$$\eta = \frac{L(\text{syndrome output for SW coding of comp-sig-map})}{L(\text{comp-sig-map})}, \tag{9}$$

where L denotes the bitstream length. As syndrome bits are output progressively, the compression fraction η is monitored so that when it reaches a certain value η_B, SW coding is bypassed and conventional coding ensues. This value η_B, called the boundary fraction, signifies when the prog-sig-map bitstream has the same length as the arithmetic coded conventional sig-map. Therefore, η_B represents the largest fraction that maintains a reduced bitstream length after the SW coding of comp-sig-map. In this way, the performance of the proposed system is never worse than the performance of the corresponding SPC with the conventional sig-map.

Here we intend to use a predetermined boundary fraction η_B found by offline experiments, so as to obviate the continuous monitoring of η while encoding. In order to determine η_B, we encoded two images through different last thresholds. For all thresholds in both sequences, the boundary fraction is very close to 0.9. Therefore, we use $\eta_B = 0.9$ as the SW bypass condition in all our following encoding simulations. Allowing this 10% reduction of comp-sig-map bits compensates for the 3-5% penalty introduced by fixed-length coding. When the average compression fraction of the last J (an empirical $J = 4$ in our simulation) LDPC words exceeds η_B, the following coding of the current frame will bypass the SW module and be conventionally coded.

4.5. SW Codec for Value Information

The SW encoding of value information V is quite similar to that of comp-sig-map, including the LDPCA encoder and buffer. Instead of the parity generator for comp-sig-map M, the 8-bit cyclic redundancy check (CRC) [28] calculation is applied to each value codeword.

The decoding of value information is performed after the comp-sig-map decoding. The decoded prog-sig-map (S and $\hat{M}$) is applied to the reference frame Z to extract the signs and refinement bits, forming the SI value information V_{SI}, which is used by SW decoder. The SW decoder includes an LDPCA decoder and an 8-bit CRC check. The LDPCA decoder is similar to the one for comp-sig-map SW decoding. In our simulation, the intrinsic LLR's are calculated based on a fixed crossover probability $p = 0.1$, for all code-bits in all codewords for simplicity. A better-estimated p will result in better performance, which we leave for future work. After each LDPC decoding, the LDPC syndrome and the CRC-8 summations are checked to decide a successful or failed decoding, whereupon a signal is sent back to the encoder asking for more syndrome bits or asking for the syndrome of the next codeword. The SW coding is bypassed when the average SW compression fraction of the last 3 LDPC codewords exceeds 0.95.

4.6. Reconstruction

The recovered $\hat{M}$ and $\hat{V}$ are sent to the normal set partition decoder, reconstructing the SW frame $\hat{X}$, together with the sum-sig-map S. If the $\hat{M}$ and $\hat{V}$ are exactly the same as M and V, the recovered $\hat{X}$ will be the same as the recovered frame X^* from conventional set partition coding (in the intra coding style) with the same last coded bit-plane. Even if there are some bit errors in the $\hat{M}$, the $\hat{M}$ errors in a set (tree/block) will result in the recovery of some significance values to erroneous coordinates within that specific set, but will not impact the recovery of other sets, analogous to the locally damaging error pattern [20]. In this way, an erroneous SW decoding would not corrupt the whole following decoding process.

5. Simulation Results

Recall that the purpose of distributed source coding is to provide a low complexity encoder at the expense of additional complexity in the decoder and to achieve performance comparable to the conventional high complexity encoder and lower complexity decoder. The objectives of the following simulations are two-fold:

1. to prove that our method using the progressive significance map achieves results that are superior to the parallel distibuted method using the conventional significance map and to the non-distributed all-intra method of the same coding algorithm;
2. to show that our method achieves results that are close in performance or comparable to the best high complexity distributed and non-distributed video coding methods.

Two-dimensional SPIHT is employed as the set partition coding method in the simulations. The 3-level DWT with Daubechies' 9/7 filters [29] is applied to the frames. The channel code used for the Slepian-Wolf codec is an irregular LDPCA code of length 396 [26]. We compare our proposed method with the following five codecs.

1. The all-intra 2D SPIHT, the origin and basis of the poposed DVC system.
2. SPIHT DVC system with SW coding of value (sign and refinement) bits and non-SW coded conventional significance map [15].
3. H.264/AVC all-intra coding is the well known DCT-based intra-frame codec. It exploits the spatial correlation efficiently at the cost of some computational complexity.
4. H.264/AVC IBI is an inter coding method with no motion estimation/compensation. It thus has a lower encoding complexity compared to the full H.264/AVC inter-frame codec since it uses the co-located blocks in the previous and/or future reference frames for prediction. It is often used as a comparative method in DVC simulations [25, 18].
5. DISCOVER [25] is a well developed DCT-based DVC codec with LDPCA-based SW coding. Several coding modules in DISCOVER are much more mutual and efficient than the proposed DVC codec, such as the reference frame generation, side information extraction, virtual channel model, and soft input calculation.

The intention of the first two items in the list of methods fulfills the first objective. The intention of the next three items fulfills the second objective of the simulations. The proposed DVC codec is evaluated for the SW frames only, because we assume that the key frames are available losslessly at the decoder. Thus the reported rate/distortion does not include the key frames.

Three test QCIF (176×144) sequences are chosen for the simulations: *hall monitor* with very low motion, *foreman* with moderate motion, and *soccer* with high motion content. Each sequence consists of 150 luminance frames at 15 Hz.

Figure 2 provides the rate-distortion performance, measured in PSNR versus bit rate, of the six codecs on the three test sequences. Notice that the first objective is fulfilled, since the proposed coder beats the value-only distributed SPIHT coder, which in turn beats the all-intra SPIHT, for all tested sequences. In fact, the value-only SPIHT lies close to midway in performance between all-intra SPIHT and the proposed coder. The points on these graphs were obtained by decoding the SPIHT bitstreams to the end of selected bitplanes. That equalizes the distortion for the same bitplane for a given source. Taking the three test images together, at PSNR's between 34 and 35 dB, the rate gap between the proposed and all-intra SPIHT coder shrinks from 124 to 53 to 8 kbps, from the low motion *hall monitor* to moderate motion *foreman* to high motion *soccer*. The corresponding rate savings percentage-wise are 40, 18, and 4 %. The corresponding gaps in rate between value-only and all-intra SPIHT are 58, 24, and 5 kbps with respective rate savings of 24, 9.3, and 2.6 %.

Regarding the second objective, the comparative performance of the proposed coder varies depending on the degree of motion. In the case of *hall monitor*, the proposed method improves the PSNR performance by about 2 dB and 4 dB compared to the two intra methods, H.264 intra and 2D SPIHT, respectively, but it is not as good as the DISCOVER DVC system. For *foreman* the proposed method has similar performance with DISCOVER and H.264 intra, and is more than 1 dB better than 2D SPIHT. The performance of the DISCOVER system is degraded significantly with *soccer*, which contains large and rapid movements. The proposed method however still provides a stable rate-distortion performance, which is a little better than 2D SPIHT, about 2 dB higher than DISCOVER, and about 1 dB lower than H.264 intra. The inter-frame codec (the H.264/AVC IBI) outperforms the other methods for all sequences. The following observations may also be gleaned.

- The performance of the distributed systems (DISCOVER and the proposed system) decreases as the video movement level increases (from hall monitor to foreman and soccer), which is obvious when comparing with the H.264 intra coding curves. This can be explained by the accuracy of the reference frame. For video with low movements, higher correlated side information can be extracted from a more accurate reference frame, and consequently better compression can be achieved in the distributed way.
- As the motion decreases (from soccer to foreman and hall monitor), the performance of DISCOVER improves significantly, because of its effective reference-frame-generation approach. The proposed system adopts a very simple method (the average of two key frames) to produce its reference frame, which is of lower accuracy. With an improved estimation on reference frame, the proposed system can be expected to be more efficient. This is left for the future work.

6. Conclusion

The proposed DVC system takes advantage of the prog-sig-map and thus enables both the fixed-length-coded comp-sig-map and the value information of SPC to be SW coded. To the best of our knowledge, this is the first time that part of the sig-map can be SW coded and is simultaneously robust to decoding errors, thanks to the fixed-length feature and the consequent error resilience feature of the comp-sig-map. Simulation results show that the proposed DVC system may significantly improve the rate-distortion performance compared with its counterparts of SW coding of the value bits in the conventional sig-map and all-intra SPIHT coding and is competitive with the state-of-the-art DVC codec—DISCOVER.

It is known that the accuracy of the reference frame influences significantly the rate-distortion performance, because better correlated SI can be extracted. For the proposed DVC codec, a better rate-distortion performance can be expected, with an effective reference frame generation module. The proposed SW coding of the prog-sig-map is also suitable for some state-of-the-art DWT-based DVC codecs [18, 30], where the effective source classification, advanced temporal interpolation, iterative MMSE estimation, etc. techniques make their performance somewhere between H.264/AVC all-intra and H.264/AVC IBI. However, their position-by-position sig-map is still very sensitive to decoding errors. The proposed scheme can alleviate this problem. At the same time, further compression can be expected, due to the beneficial SPC (which was abandoned in [18, 30]) and the SW coding of the comp-sig-map, especially when there is highly correlated side information.

References

[1] ISO/IEC, Information technology—JPEG 2000 image coding system, Part-1: Core coding system, ISO/IEC 15444-1.

[2] ISO/IEC, Information technology—JPEG 2000 image coding system, Part-2: Core coding system, ISO/IEC 15444-2.

[3] CCSDS, Image data compression, CCSDS-122.0-B-1 Blue Book.

[4] P. S. Yeh, P. Armbruster, A. Kiely, B. Masschelein, G. Moury, C. Schaefer, C. Thiebaut, The new CCSDS image compression recommendation, in: Proceedings of IEEE Aerospace Conference, 2005, pp. 4138–4145.

[5] A. Said, W. A. Pearlman, A new, fast, and efficient image codec based on set partitioning in hierarchical trees, IEEE Transactions on Circuits and Systems for Video Technology 6 (3) (1996) 243–250.

[6] W. A. Pearlman, A. Islam, N. Nagaraj, A. Said, Efficient, low-complexity image coding with a set-partitioning embedded block coder, IEEE Transactions on Circuits and Systems for Video Technology 14 (11) (2004) 1219–1235.

[7] J. Andrew, A simple and efficient hierarchical image coder, in: Proceedings of International Conference on Image Processing (ICIP), Vol. 3, 1997, pp. 658–661.

[8] C. Chrysafis, A. Said, A. Drukarev, A. Islam, W. A. Pearlman, SBHP–a low complexity wavelet coder, in: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Vol. 6, 2000, pp. 2035–2038.

[9] S. T. Hsiang, Highly scalable subband/wavelet image and video coding, Ph.D. thesis, Rensselaer Polytechnic Institute (2002).

[10] W. A. Pearlman, A. Said, Set Partition Coding: Part I of Set Partition Coding and Image Wavelet Coding Systems, Foundations and Trends in Signal Processing 2 (2) (2008) 95–180.

[11] W. A. Pearlman, A. Said, Image Wavelet Coding Systems: Part II of Set Partition Coding and Image Wavelet Coding Systems, Foundations and Trends in Signal Processing 2 (3) (2008) 181–246.

[12] W. A. Pearlman, A. Said, Digital Signal Compression: Principles and Practice, Cambridge University Press, 2011.

[13] X. Guo, Y. Lu, F. Wu, W. Gao, Distributed video coding using wavelet, in: Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), 2006, pp. 5427–5430.

[14] S. A. Martucci, I. Sodagar, Zerotree entropy coding of wavelet coefficients for very low bit rate video, in: Proceedings of International Conference on Image Processing (ICIP), Vol. 1, 1996, pp. 533–536.

[15] N. M. Cheung, C. Tang, A. Ortega, C. S. Raghavendra, Efficient wavelet-based predictive Slepian-Wolf coding for hyperspectral imagery, EURASIP Journal on Signal Processing, Special Section on Distributed Source Coding 86 (11) (2006) 3180–3195.

[16] V. Thirumalai, I. Tosic, P. Frossard, Balanced distributed coding of omnidirectional images, in: Visual Communications and Image Processing 2008, Vol. Proc. SPIE 6822, 2008.

[17] Y. Zheng, A. Wang, Z. Li, Wavelet-domain distributed video coding using optimized TCQ, in: Proceedings of International Conference on

Innovative Computing, Information and Control (ICICIC), 2009, pp. 361–364.

[18] X. Li, On the importance of source classification in Wyner-Ziv video coding, in: Visual Communications and Image Processing 2008, Vol. Proceedings SPIE 6822, 2008.

[19] A. Wang, Y. Zhao, Z. Zhu, H. Wang, Scalable distributed video coding based on block SW-SPIHT, Chinese Optics Letters 5 (6) (2007) 336–339.

[20] Y. Hu, W. A. Pearlman, X. Li, Progressive significance map and its application to error resilient image transmission, IEEE Transactions on Image Processing 21 (7) (2012) 3229–3238.

[21] Y. Wang, S. Rane, P. Boufounos, A. Vetro, Distributed compression of zerotrees of wavelet coefficients, in: Proceedings of 2011 IEEE International Conference on Image Processing (ICIP), 2011, pp. 1821–1824.

[22] A. Wyner, Recent results in the shannon theory, IEEE Trans. on Inform. Thy. IT-20 (1) (1974) 2–10.

[23] D. Slepian, J. Wolf, Noiseless coding of correlated information sources, IEEE Transactions on Information Theory 19 (4) (1973) 471–480.

[24] D. Varodayan, A. Aaron, B. Girod, Rate-adaptive codes for distributed source coding, EURASIP Journal on Signal Processing, Special Section on Distributed Source Coding 86 (11) (2006) 3123–3130.

[25] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, M. Ouaret,

The DISCOVER codec: Architecture, techniques and evaluation, in: Picture Coding Symposium, 2007.

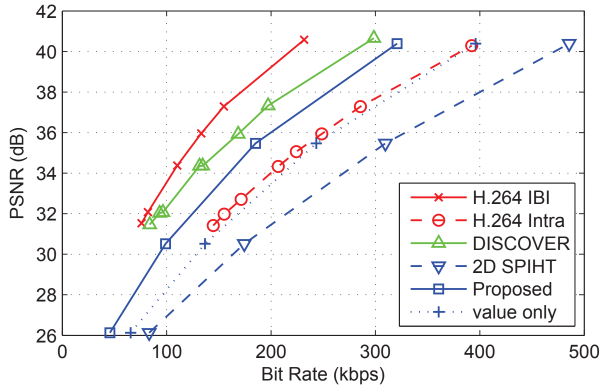
[26] LDPCA program code, http://ivms.stanford.edu/ varodayan/ldpca.html (2011).

[27] A. D. Liveris, Z. Xiong, C. N. Georghiades, Compression of binary sources with side information at the decoder using LDPC codes, IEEE Communications Letters 6 (10) (2002) 440–442.

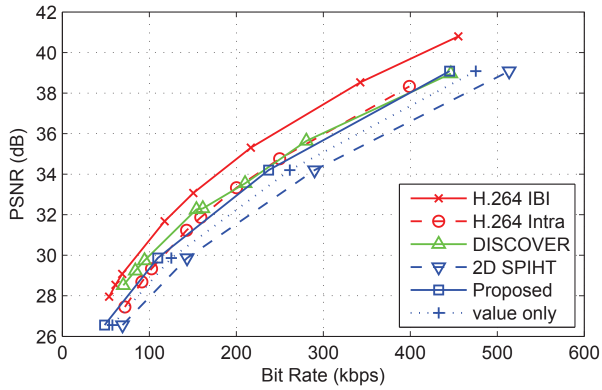
[28] W. W. Peterson, D. T. Brown, Cyclic codes for error detection, Proceedings of the IRE 49 (1) (1961) 228–235.

[29] M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, Image coding using wavelet transform, IEEE Transactions on Image Processing 1 (2) (1992) 205–220.

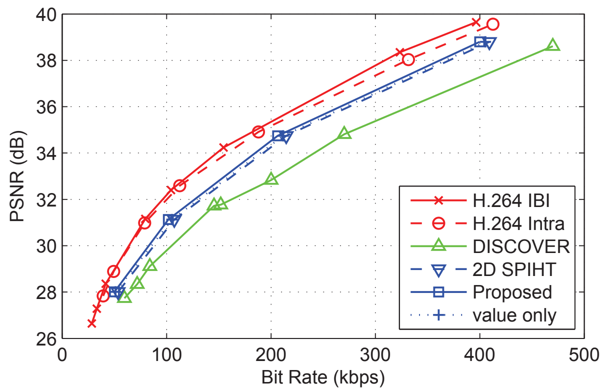
[30] X. Zong, B. Yang, A. Men, Improved side information generation and virtual channel modeling for distributed video coding, in: International Conference on Wireless Communications, Networking, and Mobile Computing, 2009, pp. 1–4.



(a) hall monitor.



(b) foreman.



(c) soccer.

Figure 2: Rate-distortion performances of test video.