EXIT Chart-based Side Information Refinement for Wyner-Ziv Video Coding

Wen Ji[1], Pascal Frossard[2], Yiqiang Chen[1]
[1] Beijing Key Laboratory of Mobile Computing and Pervasive Device,
Institute of Computing Technology, Chinese Academy of Sciences
[2]Signal Processing Laboratory (LTS4), Ecole Polytechnique Fédérale de Lausanne (EPFL)
Email: {jiwen@ict.ac.cn; pascal.frossard@epfl.ch; yqchen@ict.ac.cn}

Abstract

This paper focuses on side information (SI) refinement in Wyner-Ziv video coding and proposes to exploit the intrinsic property of channel coding for improving the joint decoding performance. In this paper, we propose to use syndrome and information bits from the encoder to help the decoder in refining the SI. We use extrinsic information transfer (EXIT) chart analysis to deduce the mutual information variation in LDPC iterative decoding during the SI refinement process. The objective is to obtain the same decoding quality under lower coding rates. Simulation results demonstrate the effectiveness of the proposed solution.

I. Introduction

Recently, Wyner-Ziv (WZ) video coding has received a large amount of research focuses because it is a practical case of distributed video coding [1] [2]. In most existing WZ video coding schemes, the video frames are grouped into key and WZ frames [3]. The key frame is similar to an Intra-frame and is encoded using conventional video coding methods. The WZ frame is based on the channel coding principles, e.g., Turbo [4] [5], low-density parity-check (LDPC) [6] [7], or Fountain [8] codes. After the transform and quantization steps, only the syndrome data is sent to the decoder. In the decoder side, the side information (SI) is created from the key frame and other decoded frames through motion-compensated interpolation/extrapolation methods. The frames are decoded by joint processing of the SI and syndrome bits from the encoder. However, WZ video coding still did not reach the compression efficiency performance of conventional video coding solutions, mainly due to the poor quality of the SI [5].

SI refinement is an important step to improve the decoding quality, which is achieved by updating and improving the SI during decoding. The SI refinement approaches can be further categorized into three classes. In the first class, SI generation and refinement mainly depend on the motion-compensated interpolation/extrapolation that is performed either in the temporal domain [9], the spatial domain [10], or the transform domain [5]. These techniques use all the available information about the frame being decoded without any extra information from encoder. However, for quick or sudden motion in the video sequence, or for the group of pictures (GOP) with long intervals, or for reference frames with large noise, the SI accuracy drops quickly, which inevitably leads to the loss of rate-distortion (R-D) efficiency.

The second class focuses on using additional information from the WZ encoder to improve the prediction accuracy and results in Hash-based motion estimation. The encoder generates hash codewords corresponding to the coordinates of original block through Hash function [11], and sends the hash

This work was supported by the National Natural Science Foundation of China (61001194), and by the Beijing Natural Science Foundation (4122078).

1068-0314/12 $26.00 © 2012 IEEE
DOI 10.1109/DCC.2012.28

codewords and the syndrome bits together to the decoder. The purpose of hash codewords is to aid the decoder in accurately estimating the motion [12]. However, it is still difficult to construct a Hash function with good collision-resistant properties.

The third class refines the SI from the syndrome information. In their seminal paper [4], Aaron and Girod proposed that lower coding rates can be achieved through puncturing the parity bits (syndrome part). This implies that the SI quality can be refined through additional parity bits subsets. The reason is that WZ video coding regards the statistical dependency between the original frame and the SI as a virtual "correlation channel" [1]. However, when SI is inaccurate and exceeds the error correction capability of the channel decoder, the WZ frame can not be successfully decoded even when more parity bits are introduced.

The above methods improve the SI accuracy from either information bits or syndrome bits. In this paper, we propose to use syndrome and information bits from the encoder to aid the decoder refining the SI. We adopt the point of view of studying the SI refinement by analyzing and optimizing the LDPC decoding process. We analyze the correlation between the refined SI and the syndrome data, which determines adequate adjustment to be performed between these data. The iterative operations of the channel decoder and the convergence of the iterative process can be characterized with an EXIT chart [13], where the change in extrinsic mutual information reflects the evolution of the correlation in the refinement steps. Based on the EXIT chart analysis, the convergence properties of the iterative decoding can thus be explicitly taken into account in the SI refinement process. The bitrate of the aid information and parity bits is optimized according to the correlation variance under convergence conditions. The decoding performance is optimized by formulating a conditional optimization problem, which is solved through an interior-point method [14]. The overall objective is to improve the WZ video coding performance through minimizing the coding bitrate. We show by experiments that the exploitation of the intrinsic properties of channel coding techniques leads to improved performance in WZ video coding. The proposed method can be extended to other SI refinement frameworks and other channel coding algorithms.

The rest of the paper is organized as follows. In Section II, we describe the system model and formulate the proposed SI refinement solution. We analyze the decoding process and write the corresponding EXIT function that is used to describe the correlation during SI refinement. In Section III, we present the optimization solution for SI refinement. Section IV provides the simulation results and Section V concludes.

II. System Structure and SI Refinement Model

A. WZ Frame Codec Structure with Proposed SI Refinement

We first illustrate in Fig. 1 the distributed video coding system considered in this paper. In the WZ frame coding, the DCT and quantization stages follow the general design of distributed video coding systems. Then, the binary source sequence is encoded as the syndrome of the Rate-Compatible LDPC (RC-LDPC) encoder. Let $\mathbf{X}^N = \{x_1, ..., x_N\}$ be the binary source sequence. The LDPC encoder, with codeword length M, is characterized by a $(M-N) \times N$ generator matrix $\mathbb{G}$. The syndrome bits are generated from $(\mathbf{Z}^{M-N})^{\mathrm{T}} = (\mathbf{X}^N)^{\mathrm{T}}\mathbb{G}$, where $(\cdot)^{\mathrm{T}}$ represents the transpose operator. With RC-LDPC, the variable code rate $\mathring{r}_c = \frac{N}{M}$ can be achieved by puncturing techniques. It generates the compatible output $\mathbf{Z}^P$, where $P \leq M - N$. The output of RC-LDPC encoder is cached into two buffers, one is for parity bits (the subset of syndrome $\mathbf{Z}^{M-N}$) $\mathbf{Z}^P = \{z_1, ..., z_P\}$ and the other is for aid information bits $\mathbf{X}^s$ (this part can also use a Hash function to further improve compressibility).

At the decoder, the subset syndrome data $\mathbf{Z}^P$ (often, $P << M - N$) from the encoder and the initial SI $\mathbf{Y}^N = \{y_1, ..., y_N\}$ is used as the input of the LDPC decoder. The initial SI is obtained from the key

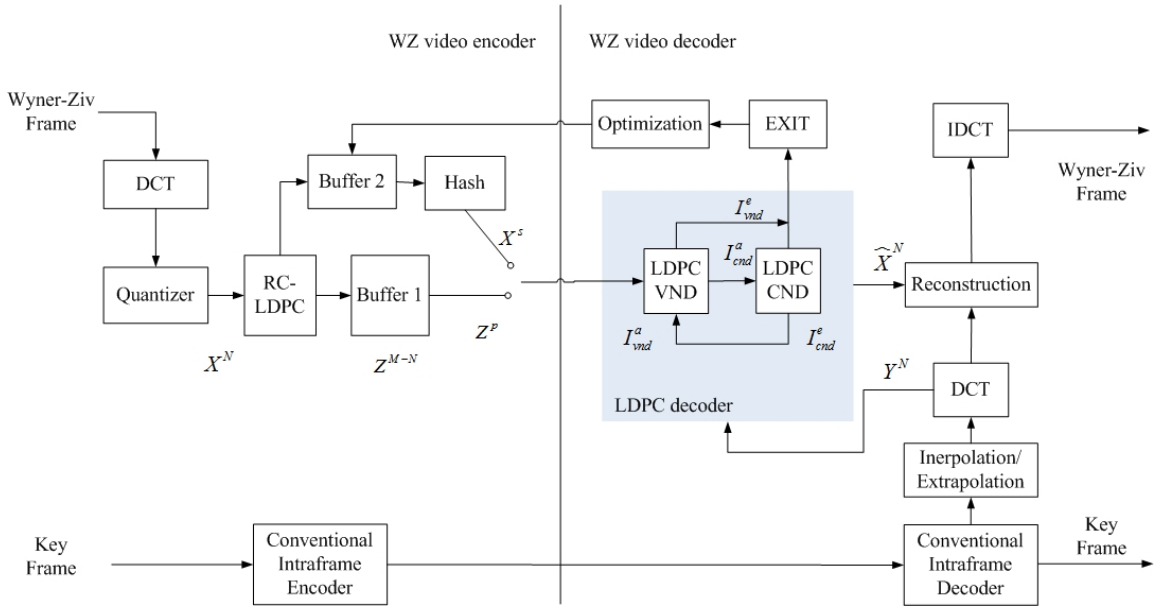

Figure 1. WZ video coding with EXIT chart-based side information refinement.

frame. Then the SI is refined along the decoding process. Specifically, refined SI is obtained from the information part by 1) decoding the current syndrome and the previous SI version; or 2) decoding the previous and current aid information and the syndrome bits from the encoder. The decoder is allowed to run a maximum of L iterations, and if convergence has not been reached, decoding is considered to have failed. In order to predict the convergence performance independently, the EXIT module is introduced through keeping track of the change in mutual information. This helps to properly adapt the aid information sent by the encoder.

B. SI Refinement Formulation

Initially, the SI $\mathbf{Y}^N$ from the key frame and the syndrome $\mathbf{Z}^P$ are both used as the inputs of LDPC decoder, the first refined SI $\mathbf{Y}^N_{(1)}$ is obtained from the information part after LDPC decoding, that is,

$$\{y_1, ..., y_N\} \times \{z_1, ..., z_P\} \longrightarrow \mathbf{Y}^N_{(1)} \tag{1}$$

The SI refinement in this work is described as follows,

Definition 1 (Information-aid SI Refinement): A refined SI codeword for the source $\mathbf{X}$ is obtained by refining the SI $\mathbf{Y}$ as follows. The encoder sends the aid information bits $\mathbf{X}^{R_s} = \{q_1, ..., q_{R_s}\}$, where R_s is the number of original WZ bits in SI, which is used to update the SI. First, $\mathbf{X}^{R_s}$ is used for updating $\mathbf{Y}^N_{(1)}$ into $\hat{\mathbf{Y}}^N_{(1)}$; then, joint decoding of $\hat{\mathbf{Y}}^N_{(1)}$ and the original syndrome $\mathbf{Z}^P$ leads to the refined SI $\mathbf{Y}^N_{(2)}$, as follows,

$$\phi : \begin{cases} \mathbf{Y}^N_{(1)} + \mathbf{X}^{R_s} \longrightarrow \hat{\mathbf{Y}}^N_{(1)} \\ \hat{\mathbf{Y}}^N_{(1)} \times \{\mathbf{Z}^P\} \longrightarrow \mathbf{Y}^N_{(2)} \end{cases} \tag{2}$$

Definition 2 (Syndrome-aid SI Refinement): A refined SI codeword for the source $\mathbf{X}$ is obtained by refining the SI $\mathbf{Y}$ as follows. The encoder sends the aid syndrome $\mathbf{Z}^{R_c}$, where R_c is the number of syndrome bits. The original syndrome $\mathbf{Z}_P$, the aid syndrome $\mathbf{Z}^{R_c}$ and the SI $\mathbf{Y}^N_{(1)}$ are then jointly decoded

to provide refined SI $\mathbf{Y}_{(2)}^N$ as follows,

$$\psi : \mathbf{Y}_{(1)}^N \times \{\mathbf{Z}^{R_c} + \mathbf{Z}^P\} \longrightarrow \mathbf{Y}_{(2)}^N \tag{3}$$

Remark: Definitions 1 and 2 imply two refinement schemes. Since each refinement scheme leads to different performance under different correlation characteristics, we optimize both schemes according to the correlation changes analyzed in next paragraphs.

C. SI Refinement During RC-LDPC Decoding

The RC-LDPC decoder is characterized by a $P \times (N + P)$ parity check matrix $\mathbb{H}$. The output $\hat{\mathbf{X}}^N$ is decoded from the available syndrome $\mathbf{Z}^P$ and the correlated sequence, the refined SI $\mathbf{Y}_{(2)}^N$. Generally, the decoding of LDPC codes can be represented by a message passing algorithm on the bipartite graph. The message between variable and parity check nodes represents the Log Likelihood Ratio (LLR) of a symbol being 0 or 1. The SI refinement is embedded in the LDPC decoding process. The bipartite graph has $N + P$ variable nodes corresponding to the codewords $\mathbf{Y}_{(2)}^N$ and $\mathbf{Z}^P$ along with P check nodes corresponding to the codeword $\mathbf{Z}^P$. Let $\mathcal{P}(j)$ denote the set of check nodes connected to the variable node j; $\mathcal{P}(j)\backslash i$ represents the set $\mathcal{P}(j)$ excluding the i-th variable node. Accordingly, let $\mathcal{Q}(i)$ denote the set of variable nodes that participate in the i-th parity-check node; $\mathcal{Q}(i)\backslash j$ represents the set $\mathcal{Q}(i)$ excluding the j-th check node. Since the statistical dependency between the source binary sequence $\mathbf{X}$ and the SI $\mathbf{Y}$ can be regarded as a virtual correlation noise channel, and since DCT coefficients for images exhibit Laplacian distribution [15], the channel model between $\mathbf{X}$ and $\mathbf{Y}$ complies with the Laplacian distribution [16], $f_{x|y}(x) = \frac{\alpha}{2}e^{-\alpha|x-y|}$, where α is the model parameter, given by correlation noise variance $\sigma_n^2 = \frac{2}{\alpha^2}$. The LLR of a binary valued random variable can be written as

$$f_j = \ln \frac{p(u_j = 0|Y)}{p(u_j = 1|Y)} = \ln \frac{\frac{\alpha}{2}e^{-\alpha|0-y_j|}}{\frac{\alpha}{2}e^{-\alpha|1-y_j|}} = \alpha(1 - 2y_j) = \sqrt{\frac{2}{\sigma_n^2}}(1 - 2y_j) \tag{4}$$

LDPC decoding can then be performed with efficient implementations such as [17]. The decoding algorithm proceeds with the following steps:

Initialization: Each variable node j is assigned an a posteriori LLR: $L(u_j) = \log\{p(u_j = 0|y_j)/p(u_j = 1|y_j)\} = f_j$.

Iteration:

1) Check-node update: for each i and $j \in \mathcal{Q}(i)$, compute

$$\Lambda_{j\to i}(u_i) = \ln \frac{1 + \prod_{i' \in Q(i)\backslash j} \tanh\left[\lambda_{i'\to j}(u_j)\right]/2}{1 - \prod_{i' \in Q(i)\backslash j} \tanh\left[\lambda_{i'\to j}(u_j)\right]/2} \tag{5}$$

2) Variable-node update: for each j and $i \in \mathcal{P}(j)$, compute

$$\lambda_{j\to i}(u_j) = L(u_j) + \sum_{j' \in P(j)\backslash i} \Lambda_{j'\to i}(u_j) \tag{6}$$

For each j, compute

$$\lambda_j(u_j) = L(u_j) + \sum_{j \in P(j)} \Lambda_{i\to j}(u_j) \tag{7}$$

3) Decision: if $\lambda_j(u_j) \geq 0$, then $\hat{u}_j = 0$; otherwise $\hat{u}_j = 1$. If $\hat{u}\mathbb{H}_P^T = 0$, then halt; otherwise $\hat{\mathbf{U}}$ is the output of the decoder, and the N front number symbols in $\hat{\mathbf{U}}$ are the refined SI $\mathbf{Y}_{(2)}^N$. Otherwise, continue iterating. If the algorithm does not halt within a maximum of L iterations, then declare a decoder failure.

D. EXIT Function in SI Refinement through RC-LDPC Decoding

The sets of variable and check nodes in the LDPC decoder are referred as the variable-node decoder (VND) and check-node decoder (CND), respectively. Iterative decoding is performed by exchanging extrinsic information between the VND and CND [13]. The EXIT chart analysis allows to predict the convergence behavior of iterative decoding [18] by tracking the mutual information changes. During each iteration, it computes the mutual information I^a_{VND} and I^e_{VND} of the input and output of VND, which represent the mutual information changes of the variable nodes. It also computes the mutual information I^a_{CND} and I^e_{CND} of the input and output of CND, which represent the mutual information changes of the check nodes. Then, through the analysis of the relation between the curves $I^e_{\mathrm{VND}} = \hat{\mathrm{g}}(I^a_{\mathrm{VND}})$ and $I^a_{\mathrm{CND}} = \hat{\mathrm{h}}(I^e_{\mathrm{CND}})$, we can accurately determine the noise threshold σ^2_{th} of LDPC codes.

$$I^e_{\mathrm{VND}} = \sum_{i=1}^{d_v} \lambda_i \cdot J\Big[\sqrt{(i-1)(J^{-1}(I^a_{\mathrm{VND}}))^2 + \frac{4}{\sigma_n^2}}\Big] \tag{8}$$

$$I^e_{\mathrm{CND}} = 1 - \sum_{i=1}^{d_c} \rho_j \cdot J[\sqrt{j-1} \cdot J^{-1}(1 - I^a_{\mathrm{CND}})] \tag{9}$$

where the functions $J(\cdot)$ and $J^{-1}(\cdot)$ are given by [13]. *

Then, we can get the variable nodes EXIT curve $I^e_{\mathrm{VND}} = \hat{\mathrm{g}}(I^a_{\mathrm{VND}})$ and the check nodes EXIT curve $I^a_{\mathrm{CND}} = \hat{\mathrm{h}}(I^e_{\mathrm{CND}})$, which represent the mutual information convergence points and characterize the iterative decoder's operation. The iterative decoder can keep an open "convergence tunnel" when $I^e_{\mathrm{VND}} > I^a_{\mathrm{CND}}$ [18], and thus continues towards successful decoding.

For the SI refinement processes (given by Definition 1 or 2), we initially model the correlation noise between the original binary sequence and the SI by the Laplacian distribution $f_{x|y}(x) = \frac{\alpha}{2} e^{-\alpha|x-y|}$, and $\sigma_n^2 = \sigma_0^2 = \frac{2}{\alpha^2}$. Then, the two following candidate refinements can be further deduced based on the respective EXIT functions.

Corollary 1: When the encoder sends the aid information bits $\mathbf{X}^{R_s} = \{q_1, ..., q_{R_s}\}$, the information bits can directly be applied in SI refinement, the correlation between $\mathbf{X}^N$ and $\mathbf{Y}^N$ is improved, accordingly, the noise variance decreases. We denote this decrement by $\triangle\sigma_1^2$, which leads to

$$\triangle\sigma_1^2 = \frac{1}{N}[(q_1 \oplus y_1)^2 + (q_2 \oplus y_2)^2 + ... + (q_{R_s} \oplus y_{R_s})^2] \tag{10}$$

Then, we decode the updated SI and the syndrome. Let $p_e = f_c(\mathring{r}_c, \sigma_n^2 - \triangle\sigma_1^2)$, where p_e is the bit-error-

*From the appendix in [13], $J(\sigma_{ch})$ is the mutual information $I(X; L_{ch}(Y))$. The accurate expression is

$$J(\sigma_{ch}) = 1 - \int_{-\infty}^{\infty} \frac{e^{-(\xi - \sigma_{ch}^2/2)^2/2\sigma_{ch}^2}}{\sqrt{2\pi\sigma_{ch}^2}} \cdot \log_2[1 + e^{-\xi}] d\xi$$

The approximate expression is

$$J(\sigma) \approx \begin{cases} a_{J,1}\sigma^3 + b_{J,1}\sigma^2 + c_{J,1}\sigma, & 0 \le \sigma \le \sigma^* \\ 1 - e^{a_{J,2}\sigma^3 + b_{J,2}\sigma^2 + c_{J,2}\sigma + d_{J,2}}, & \sigma^* < \sigma < 10 \\ 1, & \sigma \ge 10 \end{cases}$$

where the parameters are $a_{J,1} = -0.0421061$, $b_{J,1} = 0.209252$, $c_{J,1} = -0.00640081$, $a_{J,2} = 0.00181491$, $b_{J,2} = -0.142675$, $c_{J,2} = -0.0822054$, and $d_{J,2} = 0.0549608$.

$$J^{-1}(I) \approx \begin{cases} a_{\sigma,1}I^2 + b_{\sigma,1}I + c_{\sigma,1}\sqrt{I}, & 0 \le I \le I^* \\ -a_{\sigma,2}\ln[b_{\sigma,2}(1-I)] - c_{\sigma,2}I, & I^* < I < 1 \end{cases}$$

where the parameters are $a_{\sigma,1} = 1.09542$, $b_{\sigma,1} = 0.214217$, $c_{\sigma,1} = 2.33727$, $a_{\sigma,2} = 0.706692$, $b_{\sigma,2} = 0.386013$, and $c_{\sigma,2} = -1.75017$.

rate (BER) of the LDPC decoding, and $\mathring{r}_c = N/(N+P)$ is the current compatible channel coding rate. Since it has no closed-form expression, for a given code, the empirical formula $f_c(\cdot,\cdot)$ is an alternative, which can be obtained from simulation results (details in following sections); it approximately describes the iterative decoder's error probability as a function of both coding rate and noise variance. Since both $\mathbf{Y}$ and $\hat{\mathbf{X}}$ in WZ video coding is computed on Galois field $\mathbb{GF}(2)$ for RC-LDPC decoding, the noise variance decrement is

$$\triangle\sigma_2^2 = \sigma_n^2 - \triangle\sigma_1^2 - \frac{1}{N}[f_c(\mathring{r}_c, \sigma_n^2 - \triangle\sigma_1^2) \times N] = \sigma_n^2 - \triangle\sigma_1^2 - p_e \tag{11}$$

That is, the current correlation noise variance after this decoding is

$$\breve{\sigma}_n^2 = \sigma_n^2 - \triangle\sigma_1^2 - \triangle\sigma_2^2 \tag{12}$$

Then, if $\breve{\sigma}_n^2 < \sigma_{th}^2$, no additional bits are needed because current aid information bits are sufficient for the decoding.

Corollary 2: When the encoder sends the aid syndrome $\mathbf{Z}^{R_c}$, the BER decreases after LDPC decoding, accordingly, the correlation between $\mathbf{X}^N$ and $\mathbf{Y}^N$ is improved and the noise variance decreases. In this case, $p_e = f_s(\mathring{r}_c, \sigma_n^2)$ and we have

$$\triangle\sigma^2 = \sigma_n^2 - \frac{1}{N}[f_s(\mathring{r}_c, \sigma_n^2) \times N] = \sigma_n^2 - p_e \tag{13}$$

Again, $f_s(\cdot,\cdot)$ can be processed similarly to $f_c(\cdot,\cdot)$. Accordingly, the current correlation noise variance after this decoding is

$$\breve{\sigma}_n^2 = p_e \tag{14}$$

Then, if $\breve{\sigma}_n^2 < \sigma_{th}^2$, no additional bits are needed because the current aid syndrome bits are sufficient for decoding.

III. Optimization in SI Refinement

In this section, we focus on the minimization of the bitrate of aid syndrome and information while guaranteeing the decoding performance. From the above analysis, the problem consists in minimizing the number of aid bits while maintaining the opening of the convergence tunnel. From Subsection II-D, this is equivalent to setting the condition that the output mutual information I_{VND}^e of the VND is always larger than the input mutual information I_{CND}^a of the CND. Thus, we can express the condition for an open convergence tunnel as $I_{\text{VND}}^e \geq I_{\text{CND}}^a + \varepsilon$, where ε is a small positive value, which controls the gap between the two EXIT function curves. Then, the optimization problem for minimizing the number of SI aid bits is expressed as

$$\min \quad R_s + R_c \tag{15}$$

$$\text{s.t.} \quad I_{\text{VND}}^e \geq I_{\text{CND}}^a + \varepsilon \tag{16}$$

where R_s and R_c are the size of aid information in SI refinement processes given in Definitions 1 and 2, respectively. From Eq. (9), I_{CND}^a can be rewritten as

$$I_{\text{CND}}^a = 1 - J\left[\frac{J^{-1}(1 - I_{\text{CND}}^e)}{\sum_{j=1}^{d_c} \rho_j \cdot \sqrt{j-1}}\right] \tag{17}$$

By substituting (8) and (17) in (16), we obtain

$$\sum_{i=1}^{d_v} \lambda_i \cdot J\Bigg[\sqrt{(i-1)(J^{-1}(I^a_{\mathrm{VND}}))^2 + \frac{4}{\sigma_n^2}}\Bigg] \geq 1 - J\Bigg[\frac{J^{-1}(1-I^e_{\mathrm{CND}})}{\sum_{j=1}^{d_c}\rho_j \cdot \sqrt{j-1}}\Bigg] + \varepsilon \tag{18}$$

From the dark color module in Fig.1, we start the formulation by noticing that the mutual information at the input of the VND is equal to the mutual information at the output of CND, that is, $I^a_{\mathrm{VND}} = I^e_{\mathrm{CND}}$. Let $(I^{a,\mathrm{target}}_{\mathrm{VND}}, I^{e,\mathrm{target}}_{\mathrm{CND}})$ be the target convergence point in EXIT curves, which corresponds to the noise threshold σ^2_{th} of LDPC decoder. Then, using an interior-point method in the EXIT chart-based optimization problem, the EXIT functions can be sampled at a set of discrete points $I^e_{\mathrm{CND},k} \in [0, I^{e,\mathrm{target}}_{\mathrm{CND}}]$, where $k \in [1, K]$ indexes the discrete points and $I^e_{\mathrm{CND},K} = I^{e,\mathrm{target}}_{\mathrm{CND}}$. Then, the continuous convergence condition (18) can be replaced by

$$\sum_{i=1}^{d_v} \lambda_i \cdot J\Bigg[\sqrt{(i-1)(J^{-1}(I^e_{\mathrm{CND},k}))^2 + \frac{4}{\sigma_n^2}}\Bigg] \geq 1 - J\Bigg[\frac{J^{-1}(1-I^e_{\mathrm{CND},k})}{\sum_{j=1}^{d_c}\rho_j \cdot \sqrt{j-1}}\Bigg] + \varepsilon_k, \quad k \in [1, K] \tag{19}$$

where $\varepsilon_k = \varepsilon, \forall k$. These samples are in steps of τ (e.g., $\tau = 0.001$), such that they can reflect the original shape and trajectory of the original EXIT function curve.

Since the correlation between $\mathbf{X}$ and $\mathbf{Y}$ is modeled as the input and output of a noise channel with variance σ_n^2, better side information quality denotes a lower correlation σ_n^2, which leads to a lower rate for achieving a given decoding quality. On the other hand, the RC-LDPC code is based on the fact that different bits are subject to different noise effects. Thus, each SI refinement step leads to the update of channel state σ_n^2. We design a pilot-assisted correlation channel estimation algorithm, where the pilot symbol sequence $\{WZ_1, ..., WZ_\eta\}$ is a binary sequence from η pixels (a small subset of original frame). This pilot symbol sequence is transmitted together with the syndrome and thus the resulting rate increment can be ignored. The decoder correlates the received sequence with the pilot symbol sequence and estimates the initial σ_n^2 as

$$\sigma^2_{n,0} = E\Big[(WZ_k - SI_k)^2\Big] - \Big[E(WZ_k - SI_k)\Big]^2 \tag{20}$$

where the $SI_k, k \in [1, \eta]$ represents the corresponding value in the current SI.

The rates R_s and R_c in (15) are functions of the correlation noise variance σ_n^2. The accurate expressions of $R_s(\sigma_n^2)$ and $R_c(\sigma_n^2)$, which is also the inverse functions of $f_s(\cdot,\cdot)$ and $f_c(\cdot,\cdot)$ in Corollaries 1 and 2, depend on the performance of a given LDPC codec. Their estimation is dependent on accurately predicting the convergence threshold of a code with given check and variable node degree profiles. In this work, we use a curve fitting algorithm defined by formula (21), to rapidly deduce the performance of a given LDPC decoder; that simplifies the estimations of $R_s(\sigma_n^2)$ and $R_c(\sigma_n^2)$.

$$\begin{cases} \Big\langle R_s(\sigma_n^2)\Big\rangle = \arg_{\langle \hat{R}_s, \hat{\sigma}_n^2\rangle} \inf\Big\{\Big|\hat{R}_s(\hat{\sigma}_n^2) - \sum_{i=0}^{m}\binom{m}{i}\big(\frac{\sigma^2_{n,0}-\hat{\sigma}_n^2}{\sigma^2_{n,0}}\big)^{m-i}\big(\frac{\hat{\sigma}_n^2}{\sigma^2_{n,0}}\big)^i R_{s,i}\Big|\Big\} \\ \Big\langle R_c(\sigma_n^2)\Big\rangle = \arg_{\langle \hat{R}_c, \hat{\sigma}_n^2\rangle} \inf\Big\{\Big|\hat{R}_c(\hat{\sigma}_n^2) - \sum_{i=0}^{m}\binom{m}{i}\big(\frac{\sigma^2_{n,0}-\hat{\sigma}_n^2}{\sigma^2_{n,0}}\big)^{m-i}\big(\frac{\hat{\sigma}_n^2}{\sigma^2_{n,0}}\big)^i R_{c,i}\Big|\Big\} \end{cases} \tag{21}$$

We can now rewrite the optimization of Eqs (15) and (16), by substituting Eqs (21) and (19), as

$$\min \quad \left\langle R_s(\sigma_n^2)\right\rangle + \left\langle R_c(\sigma_n^2)\right\rangle \tag{22}$$

$$\text{s.t.} \quad \sum_{i=1}^{d_v} \lambda_i \cdot J\left[\sqrt{(i-1)(J^{-1}(I^e_{\mathrm{CND},k}))^2 + \frac{4}{\sigma_n^2}}\right] \geq 1 - J\left[\frac{J^{-1}(1-I^e_{\mathrm{CND},k})}{\sum_{j=1}^{d_c}\rho_j \cdot \sqrt{j-1}}\right] + \varepsilon_k, \quad k \in [1,K] \tag{23}$$

Since the objective function (22) is a convex function of the noise variance, it can be solved efficiently by using optimization tools such as interior-point methods. Under the same SI-aid budget, a look-up table is used to compare the noise variance decrements $\breve{\sigma}_s^2$ and $\breve{\sigma}_c^2$. If $\breve{\sigma}_s^2 > \breve{\sigma}_c^2$, we select the SI-aid as information bits; otherwise, we select it as syndrome bits, until the noise threshold σ_{th}^2 is reached.

IV. Simulation results

A. EXIT Analysis Results about I^e_{VND} and I^a_{CND}

In this section, we show both the EXIT curves as well as simulation results for RC-LDPC codes. First, we implement half rate mother LDPC codes with a regular structure at block length of 8010. The degree distribution pair is $\lambda(x) = 0.5x + 0.3x^2 + 0.2x^7$ and $\rho(x) = x^6$. Then, we can accurately predict the convergence threshold under each code rate through tracking the mutual information changes. We calculate the EXIT curves using the results of (8) and (9). Fig.2 shows pairs of EXIT curves for RC-LDPC codes at rates $\mathring{r}_c$=0.5, 0.53, 0.56, respectively. The blue line is the CND $I^a_{\mathrm{CND}} = \hat{\mathrm{h}}(I^e_{\mathrm{CND}})$ curve and the red line is the VND $I^e_{\mathrm{VND}} = \hat{\mathrm{g}}(I^a_{\mathrm{VND}})$ curve under different rates. We can observe that the decoder can keep an open convergence tunnel when the VND curve is above the CND curve. The codes can converge at the predicted threshold values $E_b/N_0 = 2.8\mathrm{dB}, 3.8\mathrm{dB}, 5.2\mathrm{dB}$ for $\mathring{r}_c$=0.5, 0.53, 0.56, respectively.

B. Relation between Rates and Correlation Noise Variance

The actual relations between the rates and the correlation noise variance have close relation with the design of RC-LDPC codes. The reason is that for given rates R_c and R_s, the performance of RC-LDPC codes depends on the degree distribution of the mother code. It still has no closed-form expression. Therefore, we fit the actual curves on the RC-LDPC codes in subsection IV-A. We test two cases with $\sigma_0 = \{0.1, 0.8\}$ and $\mathring{r}_c = \{0.9, 0.8, 0.72, 0.66, 0.56, 0.53, 0.5\}$. Fig.3 plots the variance σ_n^2 as the functions of two coding bitrates R_s and R_c, respectively. The blue lines correspond to the results of the first type of refinement (Definition 1, R_s case); the red lines correspond to the second type of refinement (Definition 2, R_c case). Based on the fitting algorithm in equation (21) , the simulation results suggest that it is reasonable to model $\log(\sigma_n^2)$ with the simple relation $\log(\sigma_n^2) = \sum_{i=0}^{6} a_{s,i}(\sigma_0)R_s^i$ and $\log(\sigma_n^2) = \sum_{j=0}^{6} a_{c,j}(\sigma_0)R_c^j$, where $a_{s,i}$ and $a_{c,i}$ are fitting coefficients estimated with a Monte Carlo method. Fig.3 also demonstrates that, when the correlation between SI and WZ is weak, the information-aid SI can provide better performance alternatively, when the correlation is strong, the syndrome-aid SI provides better performance.

C. The Simulation Results in a WZ Video Coding Architecture

We implement a simple WZ video coding architecture according to the scheme in Fig.1. The key (I) frame uses JPEG coding, with peak signal-to-noise ratio (PSNR) of 37 dB or better, and the WZ frame uses RC-LDPC codec to generate syndrome bits. This structure employs the proposed SI refinement scheme, embedded in the optimization and EXIT modules. The EXIT curves of the given RC-LDPC code can be pre-computed. We use an EXIT curve look-up table to make the architecture more practical. Since the simulation mainly evaluates the performance related to bitrate, we use original information bits as information-aid SI instead of Hash codewords for fair comparisons. In fact, Hash-function results

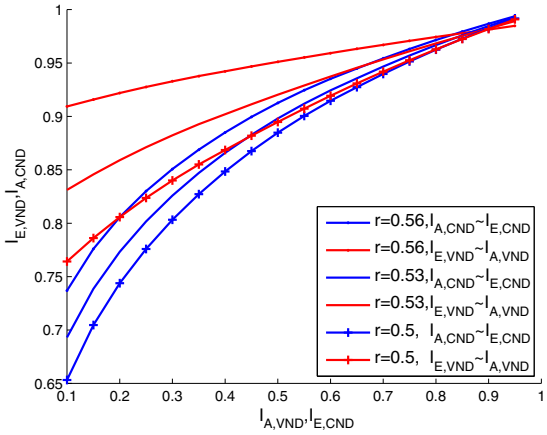

Figure 2. EXIT charts for different rate $\mathring{r}_c$.

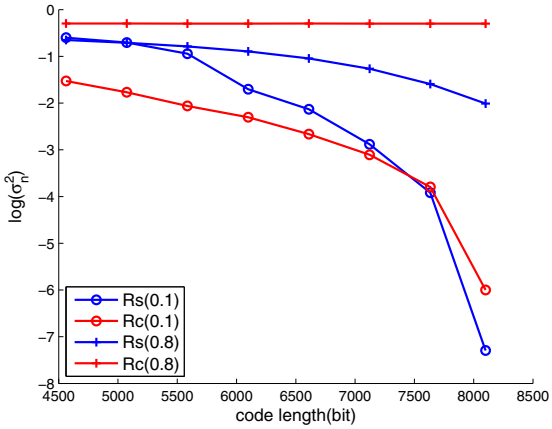

Figure 3. The relations between rates and correlation noise variance.

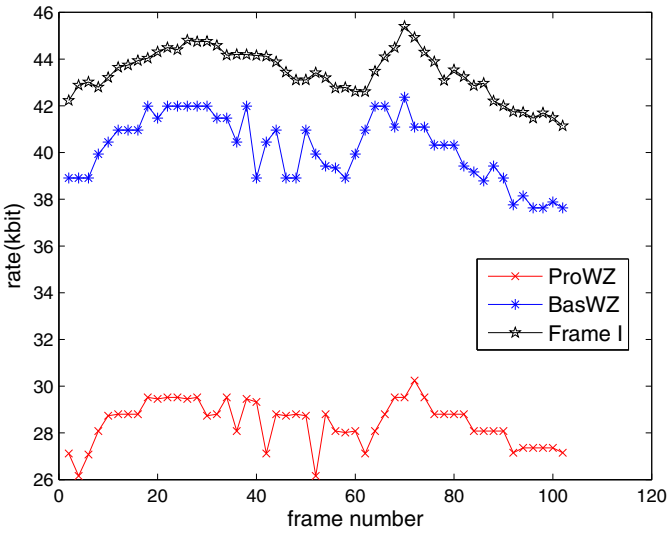

Figure 4. Frame-by-frame rate values for successful decoding.

would further improve the rate performance. We run simulations on the "Foreman" video sequence in QCIF format and encode it in I-WZ-I-WZ structure. We further compare the proposed algorithm with a baseline algorithm under RC-LDPC structure. The comparison baseline algorithm keeps sending the syndrome until LDPC decoding is successful. This is a widely used method. In the successful LDPC decoding manner, both the I and WZ frames can be reconstructed at 37dB PSNR. Fig. 4 gives the rate results for successful decoding: the black line represents the results of full I frame coding case; the blue line shows the results of the baseline algorithm; the red line illustrates the results of proposed SI refinement algorithm. The results show that the proposed SI refinement scheme obtains same coding results in lower coding rate.

V. Conclusion and Future Work

This work addresses the problem of SI refinement in WZ video coding by exploiting the intrinsic property of channel coding techniques. Simulation results show that the proposed solution can lower the WZ video coding rate. The proposed refinement solution can be extended to other SI refinement methods. In the future, we will focus on reducing the complexity of EXIT-chart computation so as to deploy this method in more practical WZ video codecs.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (61001194), and by the Beijing Natural Science Foundation (4122078).

References

[1] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proc. IEEE*, vol. 93, pp. 71–83, Jan. 2005.

[2] R. Puri, A. Majumdar, and K. Ramchandran, "PRISM: A video coding paradigm with motion estimation at the decoder," *IEEE Trans. Image Process.*, vol. 16, pp. 2436–2448, Oct. 2007.

[3] G. Esmaili and P. Cosman, "Low complexity spatio-temporal key frame encoding for wyner-ziv video coding," in *Proc. DCC*, pp. 382–390, Mar. 2009.

[4] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proc. DCC*, pp. 352–261, Mar. 2002.

[5] R. Martins, C. Brites, A. Ascenso, and F. Pereira, "Refining side information for improved transform domain wyner-ziv video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, pp. 1327–1341, Sept. 2009.

[6] D. Varodayan, D. Chen, M. Flierl, and B. Girod, "Wyner-ziv coding of video with unsupervised motion vector learning," *EURASIP J. Image Commun.*, vol. 23, pp. 369–378, Jun. 2008.

[7] K. Kasai, T. Tsujimoto, R. Matsumoto, and K. Sakaniwa, "Rate-compatible slepian-wolf coding with short non-binary LDPC codes," in *Proc. DCC*, pp. 288–296, Mar. 2010.

[8] Q. Xu, V. Stanković, and Z. Xiong, "Distributed joint source-channel coding of video using raptor codes," *IEEE J. Select. Areas Commun.*, vol. 25, pp. 851–861, May. 2007.

[9] X. Artigas and L. Torres, "Iterative generation of motion-compensated side information for distributed video coding," in *Proc. ICIP*, vol. 1, pp. I–833–6, Sept. 2005.

[10] W. Liu, L. Dong, and W. Zeng, "Motion refinement based progressive side-information estimation for wyner-ziv video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, pp. 1863–1875, Dec. 2010.

[11] T. N. Dinh, G. Lee, J.-Y. Chang, and H.-J. Cho, "Side information generation using extra information in distributed video coding," in *Proc. ISSPIT*, vol. 1, pp. 138–143, Dec. 2007.

[12] A. Aaron, S. Rane, and B. Girod, "Wyner-ziv video coding with hash based motion compensation at the receiver," in *Proc. ICIP*, vol. 5, pp. 3097–3100, Oct. 2004.

[13] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, pp. 670–678, Apr. 2004.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[15] E. Y. Lam and J. W. Goodman, "A mathematical analysis of the DCT coefficient distributions for images," *IEEE Trans. Image Process.*, vol. 9, pp. 1661–1666, Oct. 2000.

[16] C. Brites, J. Ascenso, and F. Pereira, "Studying temporal correlation noise modeling for pixel based wyner-ziv video coding," in *Proc. ICIP*, pp. 273–276, Oct. 2006.

[17] X.-Y. Hu, E. Eleftheriou, D.-M. Arnold, and A. Dholakia, "Efficient implementations of the sum-product algorithm for decoding LDPC codes," in *Proc. GLOBECOM*, vol. 2, pp. 1036–1036E, Nov. 2001.

[18] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: Model and erasure channel properties," *IEEE Trans. Inf. Theory*, vol. 50, pp. 2657–2673, Nov. 2004.