IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 11, 2021, accepted September 23, 2021, date of publication September 24, 2021,
date of current version October 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3115671

Low-Power Cross-Layer Error Management Using MIMO-LDPC Iterative Decoding for Video Processing

YOON SEOK YANG[1], (Member, IEEE), AND YONGTAE KIM[2], (Member, IEEE)
[1]Intel Labs, Intel Corporation, Santa Clara, CA 95054, USA
[2]School of Computer Science and Engineering, Kyungpook National University, Buk-gu, Daegu 41566, South Korea

Corresponding author: Yongtae Kim (yongtae@knu.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIT) under Grant NRF-2020R1A4A1019628, and in part by the Ministry of Education under Grant NRF-2019R1I1A3A01061266.

ABSTRACT This paper presents a novel low-power joint decoding system composed of a multiple-input multiple-output (MIMO) detector, low-density parity-check (LDPC) channel decoder, and H.264 source decoder for energy-aware and cross-layer source-channel transmission systems. Our aim is to achieve a minimum power design that meets the expected quality of service (QoS) at the application layer. This design includes an iterative MIMO-LDPC decoder for minimizing communication errors and an unequal error protection scheme for reducing energy consumption in transmission. The experiment results reveal that our proposed iterative MIMO-LDPC-H.264 joint decoder achieved up to 66% energy reduction with only 0.09 dB peak signal-to-noise ratio degradation when compared to a non-UEP based joint source-channel decoding system.

INDEX TERMS MIMO, LDPC, iterative decoding, unequal error protection, H.264/H.265 data partitioning, joint source-channel decoding, low-power design, error resilience, energy-aware communication system, QoS for video processing.

I. INTRODUCTION

With the increasingly widespread use of mobile systems for multimedia content production and consumption, wireless real-time video streaming and decoding in embedded systems have become important topics. However, there are many technical challenges in reliably providing high-quality video streaming in extremely low-power devices, where the power source is limited. Consequently, communication losses due to noisy channel conditions in wireless networks must be addressed. The reliability of the communication scheme as well as the resilience of the hardware to detect, decode, and process video data must be considered in an integrated fashion, including quality of service (QoS), energy, error correction, and performance.

To provide reliability and robustness to vulnerability in wireless video embedded systems, it is necessary to combine cross-layer applications and techniques for constructing communication and video processing components in the systems [1], [2]. This can provide a solution to the conflicting requirements of achieving high fidelity for video transmission with limited and varying channel bandwidth in embedded systems. To mitigate the bandwidth limitation and support a variety of applications with different QoS requirements, loss-tolerant and resilient source coding over a high-throughput wireless system can be employed. Our embedded system solution is based on a multiple-input multiple-output (MIMO) wireless system, which offers higher throughput than a single-input and single-output (SISO) wireless system [3]–[7]. The authors proposed a cross-layer resource allocation methodology to leverage video synthesizing schemes. This was devised for efficient multi-view video streaming with MIMO [4]. The authors provided insight into video transmission technologies for Quality of Experience (QoE)-aware salable video coding (SVC) over MIMO in cross-layer wireless systems [6]. Furthermore, MIMO is used in conjunction with low-density parity-check (LDPC) coding for channel coding to improve the robustness and reliability of the wireless communication [8].

The ability of a MIMO detector and LDPC decoder to suppress errors is dependent on their control parameters. For a given transmission environment, the detector and channel

The associate editor coordinating the review of this manuscript and approving it for publication was Gulistan Raja.

decoder can use more resources to enhance the bit-error-rate (BER) performance. However, less detection and decoding effort is required to reliably extract data in a high signal-to-noise-ratio (SNR) environment. Thus, a video decoder does not require the same reliability for all data blocks when a variable or assignable quality system is employed. This principle is the motivation for employing an unequal error protection (UEP) method [9]–[11] for a dynamic system that balances performance and power among the subcomponents.

Joint video specification for high-efficiency video coding (H.265/HEVC) and advanced video coding (H.264/AVC) provides error resilience features for UEP [9], [12], [13]. In particular, H.264/AVC supports flexible macroblock ordering, redundant slices, parameter set sharing, error concealment, and data partitioning (DP) techniques [10] for UEP. Thomos *et al.* reported that the use of flexible macroblock ordering associated with a UEP scheme outperforms classical H.264/AVC transmission systems with regard to the quality of reconstructed video frames [14]. In addition, the utilization of DP in H.264/AVC can yield a lower percentage of entirely lost frames [15]. A UEP scheme reprioritizes data information, aiming to reduce errors in more critical data. This is in contrast to an equal error protection (EEP) method, which does not use prioritized data information and treats all received frames equally [16].

In this paper, we propose a unified architecture and design methodology that maximizes the energy utilization for a MIMO-LDPC video decoding system. The key attribute of this proposal is that a larger overall gain can be attained if a trade-off is established for detecting and decoding for lower-priority partitions in video data. This has been demonstrated in earlier studies involving the integration of low-power and error-tolerant MIMO video and LDPC video decoders [17]–[20]. Our unified approach is based on these studies that strike a balance between power efficiency and QoS in video processing in a mobile scenario. Our approach performs an empirical analysis, searching for the minimum number of iterations of MIMO detection and LDPC decoding with regard to power savings. Energy reduction is achieved by providing a trade-off between the search space of outer iterations and inner iterations in MIMO-LDPC iterative decoding and the BER associated with the iterative decoder. A reduced number of outer and inner iterations lead to a reduction in energy consumption; however, they also lead to degradation in BER performance. To compensate for the BER degradation, we adopt the UEP scheme using the H.264/AVC DP method, which partitions the image streams into three priority groups. Figure 1 presents an overview of the unified architecture. A video source is encoded by H.264 and LDPC encoders and modulated by a MIMO modulator. Data transmitted over an error-prone channel are received and decoded by a MIMO-LDPC iterative decoder and an H.264 decoder. A trade-off occurs in the UEP-based MIMO-LDPC-H.264 decoder between decoding reliability and power consumption according to the priority of the partitioned data.

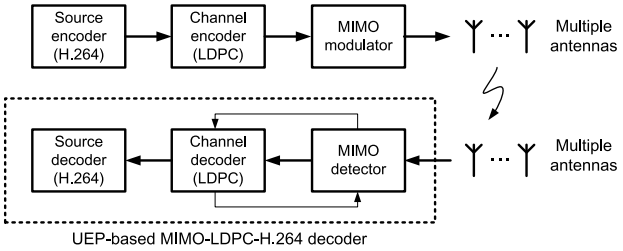

FIGURE 1. Joint decoding system.

In summary, the key contributions of this study are as follows:

- We empirically quantified the relationship between the priorities of video data and the outer and inner iterations for MIMO detection and LDPC decoding.
- We studied a novel search method for identifying the least energy-consuming sets (i.e., sets of outer iterations and inner iterations for the MIMO-LDPC decoder) to decode prioritized video data.
- We cataloged the searched iterations into a UEP lookup table and use them for the joint MIMO-LDPC-H.264 decoding at runtime.

The remainder of this paper is organized as follows. Section II briefly reviews related work, while Section III describes the MIMO-LDPC iterative decoding and H.264 DP method for UEP. Section IV presents the proposed low-power MIMO-LDPC-H.264 joint decoding scheme and design. Section V presents the simulation results and performance evaluations and Section VI summarizes and concludes this paper.

II. RELATED WORK

There have been several studies on UEP for video transmission over noisy channels to protect important partitions of video information [21], [22]. Huang *et al.* proposed a multilevel UEP by a superposition transmission coding scheme applied to H.265 DP [21]. In addition, Paudel *et al.* proposed two-level and three-level UEP algorithms, in which two-level UEP prioritized tile data based on motion density, while three-level UEP improved the protection of low-importance tiles [22]. Zhang *et al.* presented a UEP technique employing a hybrid automatic repeat request over wireless video transmission focused on energy optimization [23].

UEP approaches have been investigated to reduce the BER and energy consumption in joint decoding strategies. Hosany *et al.* proposed a MIMO-UEP approach to achieve coding gains through hierarchically modulated symbols and diversity. They applied their scheme to MIMO systems with orthogonal space-time block codes and hierarchical 4/M-ary quadrature amplitude modulation [24]. Similarly, Park et al proposed a UEP technique over MIMO orthogonal frequency-division multiplexing systems [25] and Zhang *et al.* implemented a MIMO transceiver with multi-set space-time shift keying and adopted a UEP scheme for adaptive layered video streaming for unmanned aerial

vehicles [26]. Yang *et al.* proposed a hybrid MIMO system, that consisted of spatial multiplexing for low-priority data and spatial diversity for high-priority data to achieve improved performance in terms of the BER and peak signal-to-noise ratio (PSNR) [27]. Liu *et al.* similarly divided H.264 information into two parts according to priority [28]. Li *et al.* utilized two modes: the transmission diversity mode for high error protection and the spatial multiplexing mode for a high data rate [29]. They used an unequal number of information bits in their channel error correction codes. The aforementioned studies demonstrated that compared to EEP, MIMO with UEP improves not only the capacity of the system, but also the error resilience, and overcomes frequency-selective effects of broadband wireless channels.

Joint source-channel decoding (JSCD) methods, which combine LDPC channel coding and H.264 source coding for UEP, have been proposed in several studies. [30]–[32]. Two studies proposed LDPC-based UEP algorithms using DP [30], [31]. The principle of LDPC-based UEP is that high-priority data are assigned a low code rate, whereas low-priority data are assigned a high code rate to protect relevant partitions from channel errors. Qi *et al.* developed a dynamic rate selection forward error correction scheme utilizing LDPC codes and Reed-Solomon codes for robust video communication [32].

The aforementioned studies focused on improving the received data quality or robustness of transmission using UEP. In contrast, other studies considered minimizing both the processing power for JSCD and transmission energy for the constrained video quality with Reed-Solomon channel coding [33]–[35]. In another study, Eisenberg *et al.* presented an unequal iterative decoding approach for reducing the power consumption of a channel decoder with DP and turbo decoding [36]. In this approach, more iterations are used in turbo decoding for the protection of high-priority data and minimization of the receiver power while satisfying distortion constraints specified by the video decoder at a given channel rate. Another method to reduce the power consumption of LDPC decoding was presented by Dielissen *et al.* [37]. This method exploits scalable sub-block parallelism to achieve an efficient LDPC decoding implementation for DVB-S2, enabling a lower operating frequency by reducing the parallelism of the LDPC decoder instead of using UEP. However, scalable parallelism cannot change adaptively according to a trade-off between the decoded data quality and low-power requirements. Another study proposed a JSCD scheme, that employs machine learning for UEP-based video transmission over wireless channels [38]. In the paper, the parameters of the wireless channel and influencing factors were employed to reduce noise, and the fuzzy c-means clustering algorithm was then applied for the classification of the channel environment.

III. MIMO-LDPC ITERATIVE DECODING AND UEP

Owing to the benefits of LDPC coding and the low error floor with low hardware complexity, the combination of MIMO and LDPC has been widely used in many new standards and applications [39]–[43]. The authors [40] presented an iterative detection and decoding scheme using minimum mean square error detector and a nonbinary low-density parity-check decoder to improve the performance and simplify the detector-decoder interface. They demonstrated the proposed system for a 256-QAM 4×4 MIMO system to achieve an improved error rate with iterations. An iterative detection and decoding algorithm with a soft-input-soft-output minimum-mean-square-error receiver improving BER performance over block-fading and fast Rayleigh fading channels [41] and a joint semi-definite relaxation turbo decoder that improves performance and reduces complexity in iterative turbo processing [42] were presented. The turbo receiver solved one semi-definite relaxation problem per codeword to simplify the decoder architecture without performance degradation. An iterative nonlinear detection and decoding scheme using an algorithm of users sorting and a sorting-reduced K-best method was proposed to improve performance in a multi-user MIMO system [43]. The use of MIMO detection and LDPC decoding in an iterative fashion is known to provide benefits of up to several dB. For iterative detection and decoding, a soft-input soft-output decoder and a detector are required. This section presents the mathematical background and representation of MIMO detection and LDPC decoding as well as background knowledge of UEP for QoS.

A. MIMO DETECTION

One of the challenges in the use of MIMO as a channel interface is to detect transmitted symbols. The signal received at each antenna is correlated with signals from all transmit antennas according to the channel matrix, $\tilde{H}$. One solution is to exhaustively search all possible combination of transmitted symbols for the one that is closest to the received signal. This method can provide maximum likelihood (ML) performance. However, this solution is expensive in terms of the hardware and energy cost. Another solution is to reduce the search area with a suboptimal search method. Some detection approaches based on the latter method can provide near-ML performance with a lower cost. The conversion of the MIMO problem into a tree search is described below. The received signal can be expressed as:

$$y = \tilde{H}s + n \tag{1}$$

where $\tilde{H}$ is an M×N channel matrix, s is the transmitted signal of N×1 dimensions, and n is an M×1 dimensional vector of additive complex symmetric Gaussian noise.

To solve Equation (1), one method is to exhaustively search for all possible constellations, as previously mentioned. However, sphere decoding, as expressed in Equation (2), successively reduces the search space. Using sphere decoding, the detector must only evaluate those constellations that fit inside a sphere around the received signal:

$$Problem : d(s) = \min \|y - \tilde{H}s\|^2 : d(s) < r^2 \tag{2}$$

Then, Equation (1) becomes Equation (2). We also present QR decomposition, also known as a QR factorization to convert this problem into a tree search problem. Assuming that R represents complex numbers and r represents the radius of search sphere, the following holds:

$$\tilde{H} = QL, \quad where \ Q \in R^{M\times N}, \ L \in R^{N\times N} \tag{3}$$

$$\begin{aligned}\|y-\tilde{H}s\|^2 &= \|QQ^T(y-\tilde{H}s)\|^2 + \|(I-QQ^T)(y-\tilde{H}s)\|^2 \\ &= \|Q^Ty - Ls\|^2 + \|(I-QQ^T)y\|^2 \end{aligned} \tag{4}$$

Then, Equation (2) becomes:

$$Problem : d(s) = \min \|Q^Ty - Ls\|^2 : d(s) < r^2 \tag{5}$$

$$\widetilde{y} \triangleq Q^Ty \Rightarrow Problem : d(s) = \min \|\widetilde{y} - Ls\|^2 : d(s) < r^2 \tag{6}$$

The problem can be presented as a tree search where each branch is potentially transmitted symbol, and the values of the nodes are the sum of all distances for the combination of transmitted symbols up to that level of the tree.

B. LDPC DECODING

LDPC code is error-correcting block code originally proposed by Gallager in the 1960s and rediscovered in the late 1990s [44], [45]. In these studies, the authors described an iterative two-phase message passing algorithm that involves a check-node update and variable-node update as a two-phase schedule. This code is defined by a sparse parity-check matrix H that consists mostly of 0s [46].

There have also been studies on the early termination of a frame that cannot be decoded even if the maximum number of iterations is applied [47]–[49]. In these studies, the early termination of the iterative process was determined by checking the messages during decoding. The approach in these studies was to dynamically switch off the hardware when additional iterations did not lead to an improvement in decoding performance.

C. ITERATIVE MIMO-LDPC DECODING

The iterative MIMO-LDPC decoding process is a loop that contains both MIMO detection and LDPC decoding. Both a MIMO detector and LDPC decoder should be able to accept soft values in the design of an iterative decoder. The soft output values are used again in a loop. Hochwald *et al.* [50] presented a method to calculate the approximate log-likelihood ratio (LLR) from a list of candidates. The present paper refers to the architecture of [51] for LLR computation. To generate the list, the tree search requires the following provisions:

- The tree search retains the K-best distances at the last level.
- As a new node with a shorter distance is reached, the search radius is not necessarily reduced unless all candidates in the list have shorter distances.

The radius is not limited at the beginning. In Equation (6), r is initially infinity.

Figure 2 presents the architecture of iterative detection and decoding. The MIMO detector is a list sphere detector. For

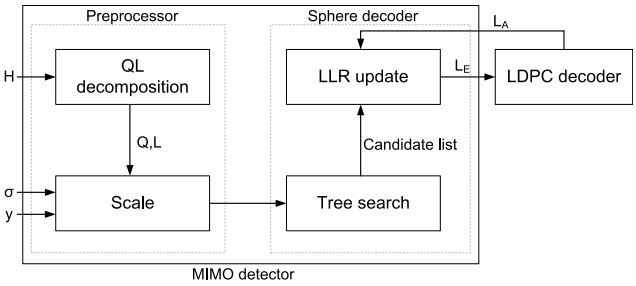

FIGURE 2. Block diagram of iterative list sphere detector.

the iterations following the first one, LLRs of the output of the LDPC decoder are fed to the LLR update block. This implies that it is not necessary to perform tree search again after the first iteration. These types of detectors search the lattice tree only once and build a list of best candidates for each received symbol. The candidate list is used to generate and update the LLRs using Equation (7) presented in [50].

$$\begin{aligned}L_E(X_k \mid Y) \approx \frac{1}{2}\max_{x\in X_{k,+1}}\{-\frac{1}{\sigma^2}\|y-Hs\|^2 + X_{[k]}^T L_{A,[k]}\} \\ -\frac{1}{2}\max_{x\in X_{k,-1}}\{\frac{1}{\sigma^2}\|y-Hs\|^2 + X_{[k]}^T L_{A,[k]}\}\end{aligned} \tag{7}$$

where $X_{[k]}^T$ and $L_{A,[k]}$ are the candidate values (-1 or 1) and LLR values, respectively, with the elimination of the kth candidate.

D. UEP SCHEME

To prevent the quality degradation of reconstructed images caused by errors, one error resilience technique used in H.264 is layered video transmission with UEP [10]. DP is one of the most widely accepted techniques for layered coding. This method make it possible to separate more important elements and less important syntax elements into different packets of data. Moreover, it enhances the application of UEP in combination with other schemes for the improvement of error/loss robustness. DP enables the partitioning of a typical slice into three parts (data partitions A, B, and C). In our UEP-based joint decoder design, we use the DP of H.264/MPEG-4 advanced video coding standard [9] developed by the joint video team of ISO/IEC and ITU-T. For simulation, we use a reference software JM14.2 [52] of H.264. This coding standard covers two layers: the video coding layer and the network abstraction layer.

We prioritize the partitioned contents into three priority groups: A, B, and C. Group A contains the most critical data (i.e., macroblock headers, motion vectors, quantization parameters, instantaneous decoding refresh picture, and parameter sets), while intra-residual data reside in group B and inter-residual data reside in group C. Each priority group contains individually coded video elements of similar importance with regard to reconstructed video quality. The prioritization process is not energy-intensive because each priority packet can be simply categorized by the content of the packet.

Therefore, we focus on energy reduction in MIMO-LDPC iterative decoding regardless of the prioritization cost.

IV. PROPOSED JOINT DECODER DESIGN

This section presents a joint decoder design using the proposed search algorithm that minimizes the number of iterations for inner and outer loops with UEP. The all-in-one joint decoding architecture optimizes the distribution of the decoding process among different application layers. The optimal utilization of MIMO and LDPC resources is analyzed and quantified by the proposed search method. In particular, the used resources in MIMO and LDPC are configured by the iteration of an outer loop and inner loop. The number of iterations of the loops performed in a search is directly related to the quality of the decoded frames and the amount of energy savings in the MIMO-LDPC iterative decoder. This section describes the proposed low-power joint decoder based on the UEP scheme.

A. LOW-POWER ITERATIVE MIMO-LDPC DECODER DESIGN

Two parameters have important effects on the performance of the iterative decoder. One is the list size, while the other is the precision of the calculations. In this subsection, we discuss the impact of these two parameters on the performance of the joint decoding system. To allow comparison with other studies, the simulation results were obtained at rate 1/2 for LDPC code and a frame size of 2304, which is the largest frame size in the WiMAX standard. The modulation was 16 quadrature amplitude modulation with a 4 $\times$ 4 antenna topology to ensure reasonable complexity. The final architecture used a list size of 32 and 8 bits for the LLRs.

1) LIST SIZE

It has been observed that the performance of this type of detector is highly sensitive to the list size [53]. A larger list size results in higher power consumption for the LLR calculation, while a smaller list results in a performance loss. Figure 3 presents the performance simulation for a list size of 2048, 1024, and 512 with 32-bit MATLAB floating point calculation precision. For each list size, different numbers of outer iterations were simulated. The performance for a list size of 1024 and 2048 was identical, which implies that increasing the list size beyond 1024 does not yield additional benefits. In [53], [54], for the K-best list sphere detector technique, the minimum list size without performance loss was 512. Consequently, the LLR update unit was the bottleneck of the system and the most power-consuming unit of the detector. The LLR update unit with a list size of 512 and throughput of 365 Mbps consumed 150.4 mW, while the remaining detector units used 22.4 mW in 45 nm technology. This level of power consumption is unacceptable for practical considerations. In [54], it was reported that with depth-first search (DFS) and proper LLR clipping, the list size could be reduced to 32. However, the DFS algorithm does not provide constant throughput like the K-best algorithm; thus, it is unsuitable for applications requiring the maximum expected completion time for each symbol.

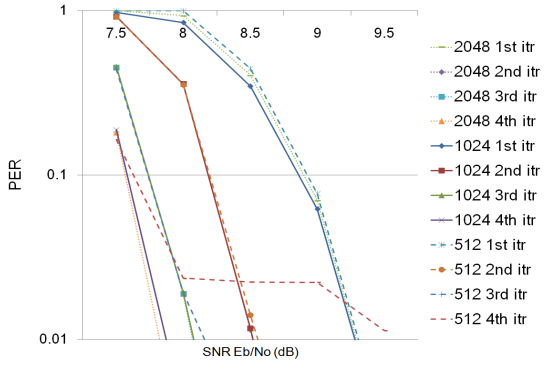

FIGURE 3. Comparison of different list sizes.

2) LLR CLIPPING

Proper LLR clipping can effectively reduce the list size. In [54], the authors concluded that an LLR clipping of eight with the DFS algorithm can provide a gain of up to 2 dB for small list sizes. This causes the estimated LLRs to be more accurate while limiting overconfident LLRs. Different list sizes and the LLR precision were evaluated in simulations, and the results are presented in Figure 4. In the figure, there is almost no performance loss for a list size of 32; in contrast, the performance degradation for smaller list sizes is considerable. Therefore, the list size of the LLR update unit and the power consumption of its implementation can be considerably reduced.

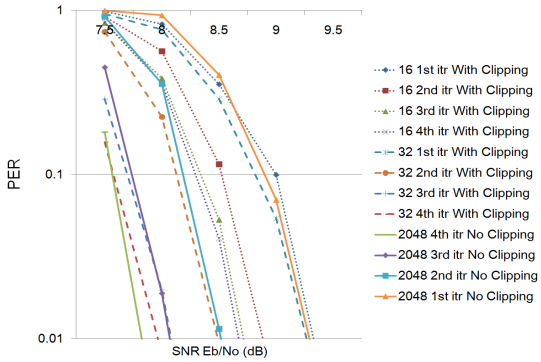

FIGURE 4. Performance for clipping.

3) LDPC DECODER DESIGN

For the iterative decoder system proposed in this paper, the LDPC decoder is implemented based on an earlier decoder presented in [55]. This decoder was developed for an IEEE 802.16e standard decoder, which uses quasi-cyclic LDPC code. Its architecture proposed on-the-fly computation for minimizing memory and recomputations by employing just-in-time scheduling. Based on this architecture, the LDPC decoder accomplished 80% savings in message passing memory requirements when compared to other semi-parallel

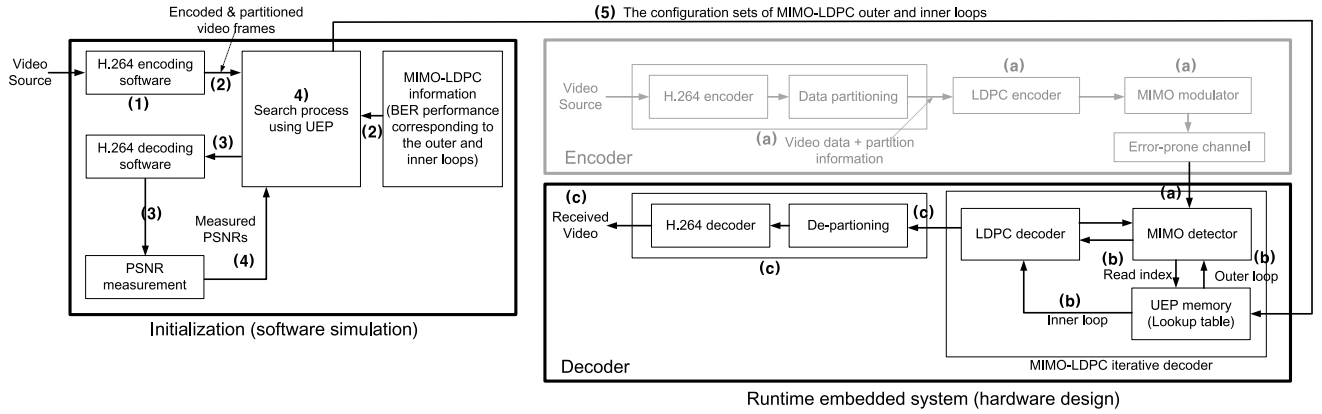

FIGURE 5. Proposed joint decoder system.

architectures based on min-sum and its variants. In particular, it proposed removal of memory needed to store the sum of the variable node messages and the channel LLRs when compared to other semi-parallel architecture. Due to the efficiency of memory utilization, the area numbers were relatively smaller than other decoders. The decoder accommodates irregular decoding with a throughput of 365 Mbps and clock frequency of 325 MHz, and uses 15 mW, in 45 nm technology to perform a maximum of 25 LDPC iterations.

4) ITERATIVE MIMO-LDPC DECODER

The LLR values at the output of LDPC decoder were fed back to the LLR update unit. The LLR values computed from a limited set of candidates create the over confidence LLR problem. LLR clipping can mitigate this problem. In the joint decoder, the LLR update unit was composed of 16 adders and calculated by Equation (8) based on the input distances and LLRs. The updated value was achieved in each clock cycle and the new LLR value was compared to the maximum LLR selected in the previous level until entries reach the last level. The update unit kept track of two numbers; LLR for the kth bit of candidate is '1' (Lambda-ML) and the other for '0' (Lambda-ML-bar). The final LLR value was updated once the LLR update unit applied the same process to all the candidates. In the LLR update block, registers (neither SRAM nor register files) were inserted to increase the throughput and to handle and store LLR values during iterative decoding process.

B. UEP-BASED JOINT DECODING PROCESS

We designed a baseline MIMO-LDPC-H.264 source-channel joint decoder based on related studies [17]–[20], [28], [53]–[55]. We devised the architecture of the proposed UEP-based joint decoder from the non-UEP baseline design. The proposed joint decoder using UEP is illustrated in Figure 5. This design mainly consists of two parts: initialization (i.e., preprocess) on a software simulation model and runtime execution on the proposed hardware joint decoder. During initialization, 1) a suite of video test streams are encoded through H.264 video encoding software [52], and 2) encoded video frames are sent to the search process with the priority information of the frames and MIMO-LDPC iteration information (BERs and the number of outer and inner iterations corresponding to each BER). 3) Channel errors are injected into the encoded frames based on the BER, and the frames are streamed to an H.264 software decoder [52], measuring the average PSNR values of the reconstructed video frames. 4) The measured PSNR values are sent back to the search process, and the sets of inner and outer iterations minimizing the power consumption are explored by the proposed search algorithm. Using this scheme, we quantify the relationship between prioritized video data and inner and outer iterations across PSNR values and find iteration sets that optimize power and performance. 5) Lastly, the searched sets are saved to UEP memory (a lookup table) for runtime execution.

At runtime, a) encoded video streams are transferred to the MIMO detector with partition information over an error-prone channel. b) The MIMO detector generates a read index corresponding to a given target PSNR, channel condition, and the priority information of input frames. This index is used as an address to read an outer and inner iteration pair from the UEP memory. This process is to be performed on-the-fly. The outer/inner loop contains outer/inner iterations. After reading an outer and inner iteration pair from the UEP memory in the joint decoder, the number of outer iterations in the iteration pair is used to select the number of back-and-forth iterations between the MIMO detector and LDPC decoder. The number of inner iterations in the pair determined the number of LDPC iterations. The output symbols of the MIMO detector are sent to the LDPC decoder, and the LDPC decoder decodes these symbols for error correction using the inner iterations. c) Frames corrected by the LDPC decoder are transferred to the H.264 video decoder to reconstruct output images. For evaluation, encoded video streams are generated by a video software encoder [52] and a MATLAB model of MIMO-LDPC encoding (see the encoder block in Figure 5). We note that the priority information (i.e. partition information) was known in each video stream and encoded by H.264 video encoder based on the quality requirement

Algorithm 1: Coarse Search Algorithm Using Binary Search

Input: PSNR_{target}

Output: The coarse searched indices i_A, i_B, i_C for data partition A, B, C while meeting the PSNR requirements.

Find i_A satisfying PSNR $(i_A, i_{max}, i_{max}) > \text{PSNR}_{target}$

Find i_B satisfying PSNR $(i_A, i_B, i_{max}) > \text{PSNR}_{target}$

Find i_C satisfying PSNR $(i_A, i_B, i_C) > \text{PSNR}_{target}$

where PSNR (i_A, i_B, i_C) denotes a PSNR result using i_A, i_B, and i_C in the iterative decoding.

Algorithm 2: Fine Search Algorithm Refining Coarse Search Results Using UEP

Input: PSNR_{target} and the coarse searched indices i_A, i_B, i_C for data partition A, B, C

Output: The fine searched indices $i_{min,A}$, $i_{min,B}$, $i_{min,C}$ for data partition A, B, C while meeting the PSNR requirements and minimum power usage.

```
// initialization
x = i_A; y = i_B; z = i_C; i_min,A = i_A; i_min,B = i_B; i_min,C = i_C; diff_max = 1;
// fine search for A
while x < i_max,A do
    x = x + 1; y = i_B;
    // fine search for B
    while y > i_min,B do
        y = y − 1; z = i_C;
        // fine search for C
        while z > i_min,C do
            z = z − 1;
            diff = (i_B − y) + (i_C − z) − (x − i_A);
            // examine requirements
            if PSNR(x, y, z) > PSNR_target && diff_max < diff then
                i_min,A = x; i_min,B = y; i_min,C = z; diff_max = diff;
            end
        end
    end
end
```

of video reconstructed images. In our experiment, it was encoded by the joint encoder as illustrated in Figure 5 and transferred to the joint decoder.

The cost of this joint decoding method is the UEP memory, requiring a 1 kbyte SRAM to store the searched sets as a lookup table. The Initialization part was performed by software simulation and the proposed UEP system used the identical video decoder, LDPC decoder, and MIMO detector that were exploited to implement the reference equal error protection and no error protection joint decoding systems for performance analysis and comparison.

The current implementation is based on a static scheme that uses presearched outer and inner iterations for MIMO-LDPC iterative decoding at runtime. However, these iterations can be reevaluated if the application or channel condition requires a change. The optimization problem of searching for the minimum number of energy iterations for MIMO-LDPC decoding can now be expressed using the prioritized video sequences, number of iterations, and PSNR values of the reconstructed video frames. The objective of our search scheme is to find a minimum set of MIMO and LDPC iterations for each priority partition while minimizing PSNR degradation. The proposed search algorithm is composed of two search steps: coarse binary search (BS) and fine search (FS). In Algorithm 1, we used binary search to find a coarse set of MIMO and LDPC iterations for priority partitions until it satisfied requirements in reconstructed image quality. The coarse searched iterations were finely tuned in Algorithm 2, searching a fine set of MIMO and LDPC iterations for the priority partitions in an exhaustive search to achieve more energy reduction while meeting the image quality requirements (i.e., desired PSNRs). We leveraged the property that increasing the number of iterations for high-importance data led to significant improvement of image quality. A small increase of inner and outer iterations for high priority partitions led to a significant decrease of iterations in low priority partitions while meeting the image quality requirements. As a result, the sets of inner and outer iterations leading to minimum energy consumption were searched by the proposed UEP algorithms.

C. COARSE SEARCH PROCESS

An outer and inner iteration pair for the MIMO detector and LDPC decoder is indexed as i. This signifies that the outer iteration for the MIMO detector and the inner iteration for the LDPC decoder are represented by an index number. For instance, if there are 32 combination pairs, index i can vary from 0 to 31, where the minimum, middle, and maximum indices $(i_{min}, i_{mid}, i_{max})$ are 0, 15, and 31, respectively. In this notation, the number of iterations for the inner loop and outer loop increases as the index increases. We observe that the BER reflects the power consumption of the iterative decoder. A high BER provides a high PSNR result and consumes more energy to process. In other words, a smaller BER leads to less power dissipation; however, it also results in a low PSNR. Therefore, we balance PSNR and power consumption.

Average PSNR values are computed by decoding the three partitions. Each data partition has a priority level and can

be decoded using different iterations to protect itself from errors. Therefore, to achieve a target PSNR, three inner and outer iteration pairs (i.e., three indices) should be determined for the three data partitions. To search for the iteration pairs that consume minimum power and satisfy QoS requirements, we use binary search [56] to find a coarse estimate represented by a set of iteration pair indices (i_A, i_B, i_C), in which i_A, i_B, and i_C denote the indices of the outer and inner iteration pairs over the priority partitions A, B, and C, respectively. The coarse set is searched when the PSNR (i_A, i_B, i_C) marginally satisfies the target PSNR (i.e., the desired quality of reconstructed images), PSNR_{target}. PSNR (i_A, i_B, i_C) represents the average PSNR of the decoded video frames at (i_A, i_B, i_C). The coarse search is detailed in Algorithm 1. In this algorithm, i_A is first searched because the average PSNR is more sensitive to errors in a high-priority partition. Therefore, when the coarse search is performed in order of importance (i.e., priority), A, B, and C, the search result quickly converges. In addition, the initial values of i_B and i_C are i_{max} while searching for i_A because the set is searched in the direction of reducing energy consumption (i.e., from a larger number of iterations to a smaller number of iterations).

D. REFINING THE COARSE SEARCH USING UEP

The FS, which refines the coarse search results, as illustrated in Algorithm 2, uses the fact that increasing the number of iterations for high-importance data leads to significant enhancement of the average PSNR. A small increment of i_A leads to a large decrement in i_B and i_C while meeting the target PSNR. By this property, we gradually increase i_A and reduce i_B and i_C until finding [$i_{min,A}$, $i_{min,B}$, $i_{min,C}$] that minimizes the energy consumption while satisfying the target PSNR.

1) Increase i_A and reduce i_B and i_C
2) Find a minimum set, [$i_{min,A}$, $i_{min,B}$, $i_{min,C}$] that maximizes the reduction of the number of indices, [($\Delta i_B + \Delta i_C$) − Δi_A], and satisfies the PSNR ($i_{min,A}$, $i_{min,B}$, $i_{min,C}$) > PSNR_{target}

$diff_{max}$ denotes the maximum reduction of the number of indices, which implies maximum energy reduction. As a result, the sets of MIMO-LDPC iterations that lead to minimum energy consumption in the joint decoding system can be searched.

V. RESULTS AND DISCUSSION

In this section, we present the performance and energy gains achieved in the UEP-based iterative decoder and discuss the benefits of the design with regard to power reduction.

A. ITERATIVE MIMO-LDPC DECODER RESULTS

The iterative MIMO-LDPC decoder was implemented and synthesized using Synopsys Design Compiler with a commercial 45nm CMOS standard cell library. The performance of the iterative MIMO-LDPC decoder with optimal parameters is summarized in Table 1. We employed 100 MHz for the operation frequency of H.264 video decoder, 284 MHz for tree search and LLR blocks in the MIMO detector, and 325 MHz for LDPC decoder generated by a phase-locked loop (PLL) in experimental implementation. These numbers were calculated to target a throughput of 365 Mbps. We plotted Figure 6 - 11, illustrating bit error rate (BER) vs. energy relations given SNR based on the implementation to compute inner and outer iteration pairs as presented in Table 2. It should be noted that the power usage of the first-iteration LLR calculation is much lower because the computation is less complex due to the absence of previous iteration LLRs. For iterations after the first one, the power usage of the detector is only due to the LLR update unit.

The relationship between the BER and energy efficiency is presented in Figures 6 - 10, where the SNRs are 7 dB, 8 dB, 8.5 dB, 9 dB and 10 dB, respectively. In these figures,

TABLE 1. Optimal parameters for different modes. *Tree search* refers to the MIMO detector, which searches for 32 best candidate symbols for each MIMO symbol, while *LLR* refers to LLR calculation, where the LLR for the received MIMO symbol is calculated based on LLRs of the candidates or previous iterations.

	Tree Search	LLR, 1st iter	LLR after 1st iter	LDPC
Freq	284 MHz	284 MHz	284 MHz	325 MHz
Throughput	365 Mbps	365 Mbps	365 Mbps	365 Mbps
Area	132 k	1.2 k	23 k	43 k
Power	22.4 mW	0.7 mW	9.6 mW	15 mW

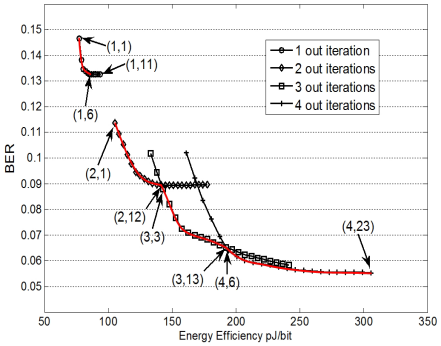

FIGURE 6. Bit error rate (BER) vs. energy, signal-to-noise ratio = 7.

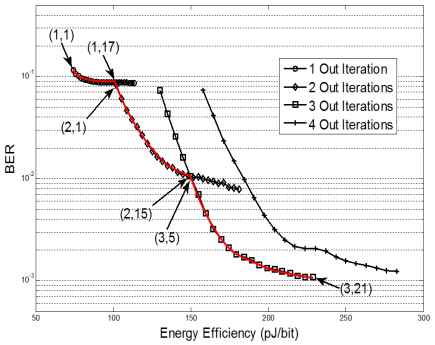

FIGURE 7. Bit error rate (BER) vs. energy, signal-to-noise ratio = 8.

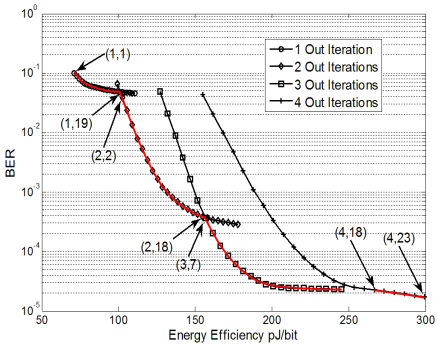

FIGURE 8. Bit error rate (BER) vs. energy, signal-to-noise ratio = 8.5.

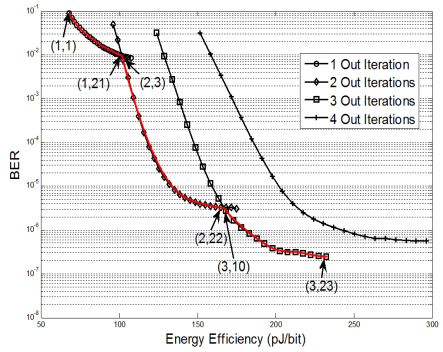

FIGURE 9. Bit error rate (BER) vs. energy, signal-to-noise ratio = 9.

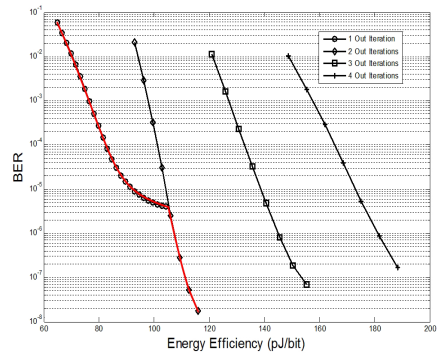

FIGURE 10. Bit error rate (BER) vs. energy, signal-to-noise ratio = 10.

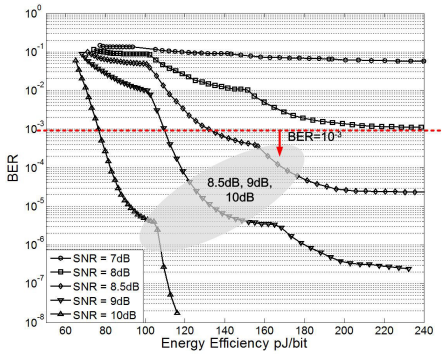

FIGURE 11. Efficiency curves at different signal-to-noise ratios (SNRs).

TABLE 2. Outer and inner iteration pairs at 8.5 dB, 9 dB and 10 dB, where i denotes the index of an iteration pair.

SNR = 8.5 dB			SNR = 9 dB			SNR = 10 dB		
i	Out	In	i	Out	In	i	Out	In
1	1	1	1	1	1	1	1	1
2	1	2	2	1	2	2	1	2
3	1	3	3	1	3	3	1	3
4	1	4	4	1	4	4	1	4
5	1	5	5	1	5	5	1	5
6	1	6	6	1	6	6	1	6
7	1	7	7	1	7	7	1	7
8	1	8	8	1	8	8	1	8
9	1	9	9	1	9	9	1	9
10	1	10	10	1	10	10	1	10
11	1	11	11	1	11	11	1	11
12	1	12	12	1	12	12	1	12
13	1	13	13	1	13	13	1	13
14	1	14	14	1	14	14	1	14
15	1	15	15	1	15	15	1	15
16	1	16	16	1	16	16	1	16
17	1	17	17	1	17	17	1	17
18	1	18	18	1	18	18	1	18
19	1	19	19	1	19	19	1	19
20	2	2	20	1	20	20	1	20
21	2	3	21	1	21	21	1	21
22	2	4	22	2	3	22	1	22
23	2	5	23	2	4	23	1	23
24	2	6	24	2	5	24	1	24
25	2	7	25	2	6	25	1	25
26	2	8	26	2	7	26	2	5
27	2	9	27	2	8	27	2	6
28	2	10	28	2	9	28	2	7
29	2	11	29	2	10	29	2	8
30	2	12	30	2	11			
...	...	...	...	...	...			
55	3	25	55	3	23			
...	...	...						
61	4	23						

each data point represents a pair of an outer iteration and inner iteration represented by (OutIter, InIter), where OutIter and InIter represent the number of outer iterations and inner iterations, respectively. In Figure 6, the four curves reveals the relationship between the BER and energy efficiency with 1, 2, 3, and 4 outer iterations, respectively. For decoding with only one outer iteration loop, the BER of the iterative decoder ranges from 0.146 to 0.133 while consuming 77.2 pJ to 90.4 pJ per bit. The point (1, 1) represents decoding with one outer iteration loop and one inner iteration loop, which achieves a BER of 0.146 and costs approximately 77.2 pj per bit. The point (1, 11) achieves a BER of 0.133 and energy efficiency of 90.4 pj per bit. However, the point (1, 6) only consumes 85.42 pj per bit while achieving the same BER performance of 0.133. In this case, the BER performance is not improved with more than six inner iterations. Thus, it is not necessary to perform more than six inner iterations with one outer iteration. As mentioned, an alternative approach for improving the BER is to increase the number of outer iterations. Decoding at point (2, 1) can achieve a BER of 0.114, which is superior to decoding with only one outer iteration loop. With more inner iterations with two outer iterations, the decoder achieves improved BER performance but increases power consumption. Thus, at the same energy consumption cost, decoding with three outer iteration loops

TABLE 3. Distributions of priority parts in test video streams, Akiyo, Foreman, Mobile, and News.

Importance	Distribution(%)			
	Akiyo	Foreman	Mobile	News
Priority$_A$	44.6%	44.8%	47.1%	38.9%
Priority$_B$	40.4%	29.6%	19.8%	34.4%
Priority$_C$	14.6%	25.4%	32.9%	26.6%

and more than three inner iteration loops can outperform decoding with two outer iteration loops with more than 12 inner iteration loops. From the perspective of energy efficiency, decoding with three outer iteration loops and more than three inner iteration loops is more cost-efficient. As the target system is energy-efficiency-oriented, the points in the lower envelope curve in Figure 6 should be selected as the optimal configuration parameters for the iterative decoder. The lower curve is marked in red in the illustration, and we refer to it as the *efficiency curve*. By this, we can derive the parameters that maximize the energy utilization.

Figures 7 - 10 present the relationship between the decoding effort and performance for different SNRs ranging from 8 dB to 10 dB, respectively. The efficiency curves are drawn by the same method used for Figure 6. Figure 11 presents all five efficiency curves for 7 dB, 8 dB, 8.5 dB, 9 dB and 10 dB SNRs. In this experiment, we aimed to keep the BER better than 10^{-3} because the H.264 decoder does not work properly with highly impaired data. The BER of the MIMO-LDPC decoder at 7 dB and 8 dB did not get better than 10^{-3}. Thus, the simulation range was for SNRs higher than 8 dB. The outer and inner iteration pairs corresponding to the BER versus energy efficiency curves at 8.5 dB, 9 dB and 10 dB, and their indices are presented in Figure 11 and Table 2, respectively. In the table, there are 61 pairs at 8.5 dB. The curve at 9 dB has the range of 1 - 55. Index i in the BER curve at 10 dB is within the range of 1 - 29, as greater than 29 produces a BER of 10^{-8}, which signifies very few errors.

B. UEP EXPERIMENT RESULTS

For the evaluation of the joint decoding system, we simulated four different quarter common intermediate format (QCIF) (176 × 144) video sources (Akiyo, Foreman, Mobile, and News: each 300 frames) at 30 fps. The distributions of the priority A, B, and C partitions in each video stream are presented in Table 3. In this table, Akiyo consists of 44.6% of A, 40.4% of B, and 14.6% of C. Mobile exhibits a relatively high distribution in priority C (32.9%), and the distribution of News is well-balanced. The amount of energy reduction can be varied according to the distribution of priority in a video sequence; therefore, the distribution information must be analyzed for the estimation of energy savings.

Table 4 presents the simulation results of the UEP-based MIMO-LDPC decoder for Foreman, Akiyo, Mobile, and News video streams. In set (x, y, z), x, y, and z denote an outer and inner iteration pair index for priority A, B, and C. The maximum indices are 61, 55, and 29 at 8.5 dB, 9 dB, and 10 dB, respectively, as presented in Table 2. Therefore, maximum PSNR values (i.e., ideal PSNR values) at 8.5 db, 9 db, and 10 dB SNRs can be achieved by (61, 61, 61), (55, 55, 55), and (29, 29, 29). The ideal PSNR values were 38.34 dB, 36.95 dB, 34.39 dB, and 37.01 dB for Akiyo, Foreman, Mobile, and News, respectively. The normalized energy reduction (ER) can be expressed by Equation (8) as follows:

$$ER_{normalized} = 1 - E_{UEP\_index}/E_{max\_index} \tag{8}$$

where E_{UEP_index} denotes the energy consumption of UEP, and E_{max_index} is the maximum energy consumption of EEP. We present the ER results for BS and BS + FS for Foreman, Mobile, and News in Table 4. As expected, the percentage of ER for BS was less than that for BS + FS, as BS is not an optimal search for minimizing resource utilization. BS + FS optimized the resource utilization with a trade-off between power and quality. This result is analyzed for Akiyo as an example.

TABLE 4. Simulation results for Foreman, Mobile, and News video streams at 8.5 dB, 9 dB, and 10 dB SNRs. The sets of iterations searched by coarse binary search (BS) and BS + fine search (FS) are presented. The percentage value of normalized energy reduction (ER) was calculated by Equation (8).

(a) SNR = 8.5 dB

PSNR$_t$	Foreman			Mobile			News		
	BS	BS + FS	ER(%)	BS	BS + FS	ER(%)	BS	BS + FS	ER(%)
30.0	(27,33,25)	(35,1,22)	66	(32,39,40)	(38,1,23)	63	(27,31,28)	(39,15,1)	67
32.0	(29,32,29)	(44,3,28)	56	(38,36,56)	(42,24,22)	50	(27,39,37)	(32,17,20)	61
34.0	(32,46,38)	(38,24,27)	50	(45,42,49)	(46,41,40)	30	(32,50,38)	(34,28,29)	50
36.0	(38,60,58)	(42,27,29)	45	(61,61,61)	(61,61,61)	0	(35,61,61)	(40,34,30)	42

(b) SNR = 9 dB

PSNR$_t$	Foreman			Mobile			News		
	BS	BS + FS	ER(%)	BS	BS + FS	ER(%)	BS	BS + FS	ER(%)
30.0	(24,24,24)	(26,2,16)	71	(25,26,27)	(28,5,8)	72	(24,26,25)	(25,24,24)	56
32.0	(24,25,25)	(26,5,23)	66	(26,26,26)	(29,11,9)	68	(24,26,25)	(25,24,24)	57
34.0	(25,25,50)	(28,13,25)	59	(29,28,27)	(29,28,27)	49	(25,25,26)	(25,25,26)	54
36.0	(26,40,35)	(27,26,25)	53	(55,55,55)	(55,55,55)	0	(26,26,48)	(27,24,25)	54

(c) SNR = 10 dB

PSNR$_t$	Foreman			Mobile			News		
	BS	BS + FS	ER(%)	BS	BS + FS	ER(%)	BS	BS + FS	ER(%)
30.0	(8,6,20)	(9,2,6)	79	(9,10,11)	(12,1,5)	77	(8,12,11)	(10,7,6)	73
32.0	(8,9,10)	(9,4,7)	76	(10,18,29)	(12,4,6)	73	(8,12,19)	(10,7,7)	72
34.0	(9,10,12)	(11,6,8)	70	(13,26,29)	(15,13,12)	53	(9,11,28)	(11,6,8)	70
36.0	(12,11,26)	(13,10,10)	61	(29,29,29)	(29,29,29)	0	(10,12,11)	(11,10,9)	65

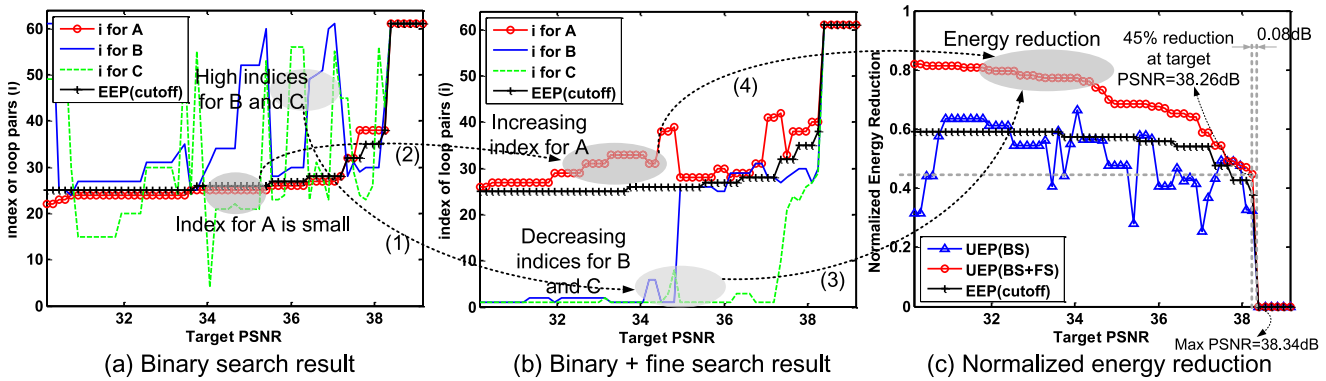

FIGURE 12. Simulation results at 8.5 dB for Akiyo: (a) coarse binary search (BS), (b) BS + find search (FS), and (c) energy reduction of proposed unequal error protection (UEP)-based energy efficient iterative decoder.

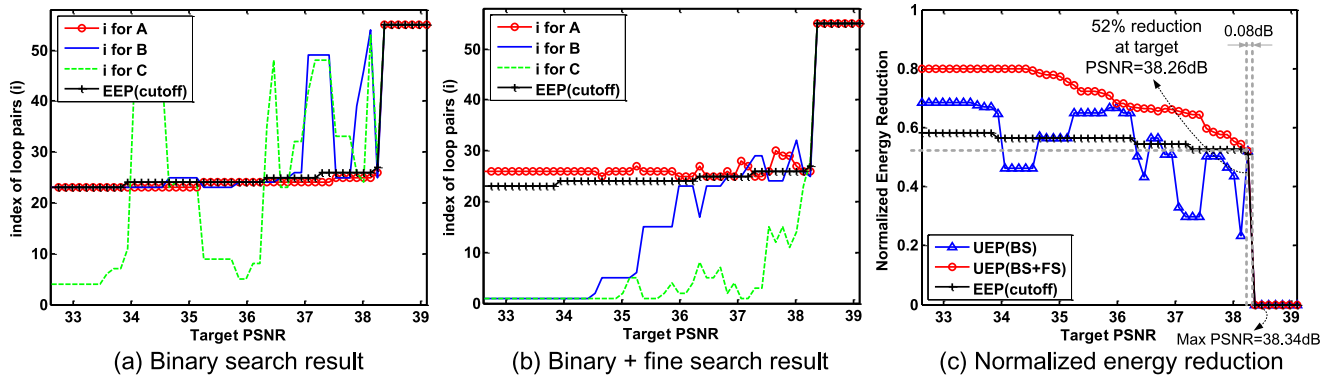

FIGURE 13. Simulation results at 9 dB for Akiyo: (a) coarse binary search (BS), (b) BS + find search (FS), and (c) energy reduction of proposed unequal error protection (UEP)-based energy efficient iterative decoder.

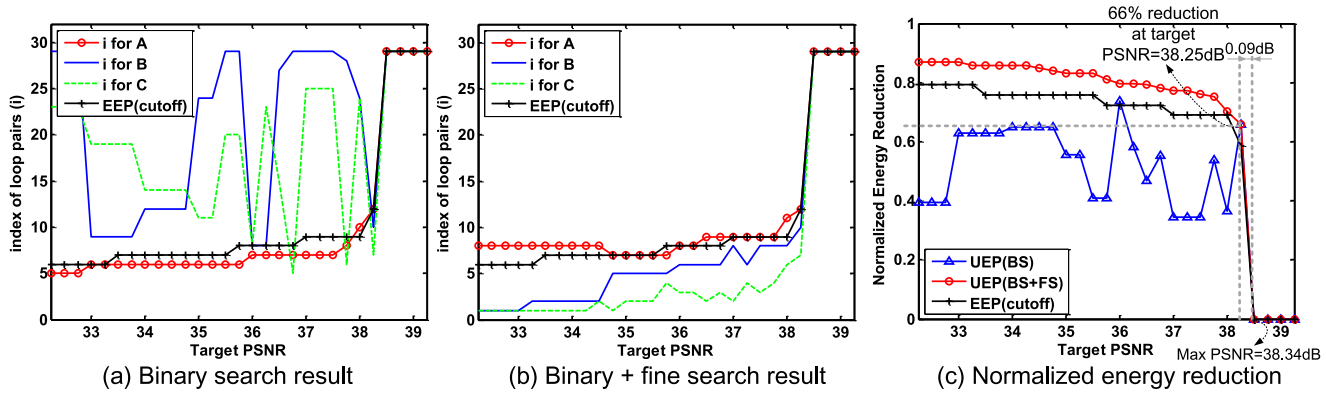

FIGURE 14. Simulation results at 10 dB for Akiyo: (a) coarse binary search (BS), (b) BS + find search (FS), and (c) energy reduction of proposed unequal error protection (UEP)-based energy efficient iterative decoder.

As also expected, the energy reduction decreased as the target PSNR increased. Moreover, we observed that the gain in energy reduction increased at high SNRs. For instance, the percentage of energy reduction at 10 dB was higher than that at 8.5 dB and 9 dB. When the target PSNR was larger than the ideal PSNR, maximum utilization of iterations occurred. Although the distributions of priority B and C on Foreman, Akiyo, and Mobile (Table 3) were different, the results at low target PSNR values such as 30.0 dB and 32.0 dB were similar to each another in terms of energy reduction. The reason was that these video streams had similar distributions in priority group A, which was dominant in performance and power consumption.

Figure 12, 13, and 14 present the experimental results for Akiyo at 8.5 dB, 9 dB, and 10 dB SNRs, respectively. In the figures, the searched indices for B and C in BS are

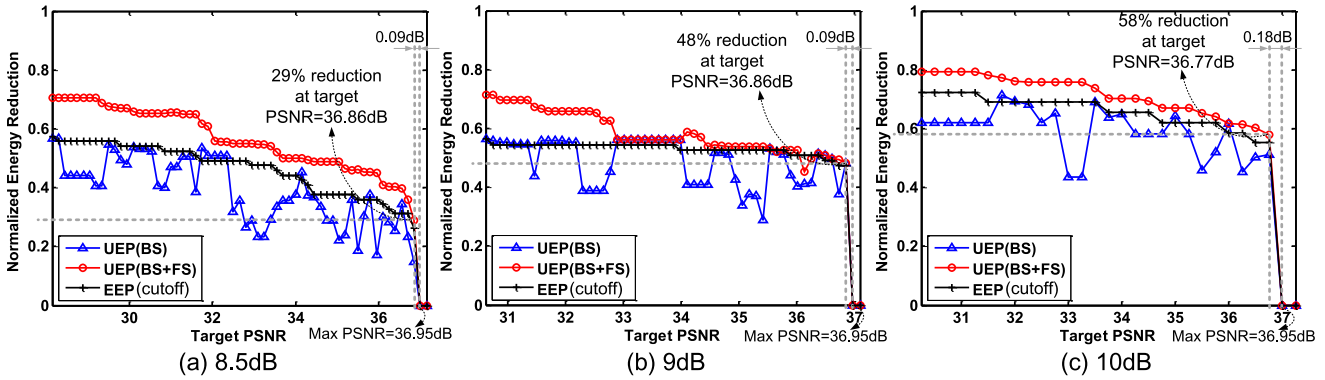

FIGURE 15. Energy reduction results for Foreman: at (a) 8.5 dB, (b) 9 dB, and (c) 10 dB.

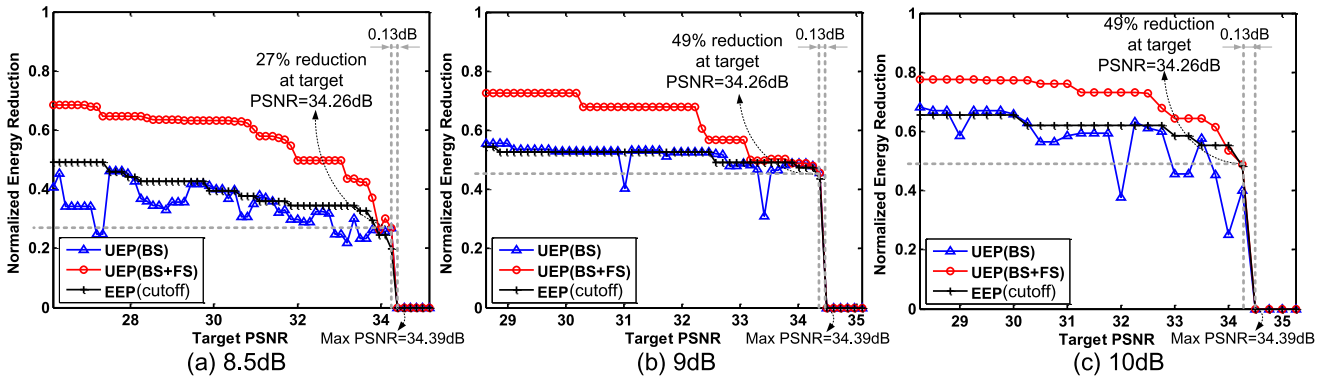

FIGURE 16. Energy reduction results for Mobile: at (a) 8.5 dB, (b) 9 dB, and (c) 10 dB.

considerably higher than the index for A. In Figure 12, the indices for B and C in BS + FS decreased significantly, as (1), with an increase in the index for A (2). This resulted in energy reduction in the iterative decoder. We thus determined the indices of EEP that equally truncates the indices for priority A, B, and C (referred to as EEP-cutoff) regardless of the priority of partitions for comparison with the UEP results. For instance, when the target PSNR was 34 dB at a 8.5 dB SNR, the set of indices achieved by UEP was (24, 30, 5) in BS, as displayed in Figure 12(a), and (34, 1, 1) in BS + FS, as displayed in Figure 12(b). The set of indices achieved by EEP-cutoff was (26, 26, 26), where all indices were the same for priority A, B, and C. In this experiment, the maximum energy reduction was achieved by UEP - BS + FS. Figure 12(c) displays the UEP and EEP-cutoff energy reduction curves versus the target PSNR values for Akiyo at 8.5 dB. These results indicate that the proposed iterative decoder yielded 45% energy reduction while while producing a negligible 0.08 dB PSNR degradation from the maximum PSNR, 38.34 dB. If the target PSNR was less than 36 dB, the decoder could reduce the energy by 70%. The EEP-cutoff results also indicate energy reduction; however, the amount was less than the UEP result. In Figures 13 and 14, the UEP results at 9 dB and 10 dB for Akiyo are similar to that of 8.5 dB. The energy reduction was 52% at the 38.26 dB target PSNR, indicating a 0.08 dB degradation from the maximum PSNR at 9 dB SNR in Figure 13(c). At 10 dB SNR in Figure 14(c), the decoder achieved 66% energy reduction at the 38.25 dB target PSNR, indicating a 0.09 dB degradation. We observed that the ER increased as the channel condition was improved. This is because a low-noise channel does not require a large number of iterations for the inner and outer loops to meet the decoding quality requirements, and the proposed UEP scheme can effectively reduce the number of iterations over a low-noise channel.

Figures 15 - 17 present the normalized energy reduction for Foreman, Mobile, and News, respectively. As observed in the experimental results for Akiyo, the normalized energy reduction for the test streams also increased as the channel condition improved. In Figure 15, the energy reduction was 29% at 8.5 dB, but improved to 48% and 58% at 9 dB and 10 dB, respectively, with negligible PSNR degradation (0.09 - 0.18 dB) for Foreman. In the Mobile results (see Figure 16), the energy reduction at 8.5 dB, 9 dB, and 10 dB was 27%, 49%, and 49%, respectively, with a 0.13 dB degradation in PSNR. In Figure 17, the energy reduction for News was 35%, 49%, and 60% at 8.5 dB, 9 dB, and 10 dB, respectively, with a 0.01 dB degradation in PSNR. Thus, the proposed joint decoding approach achieved significant power reduction without significant degradation in quality.

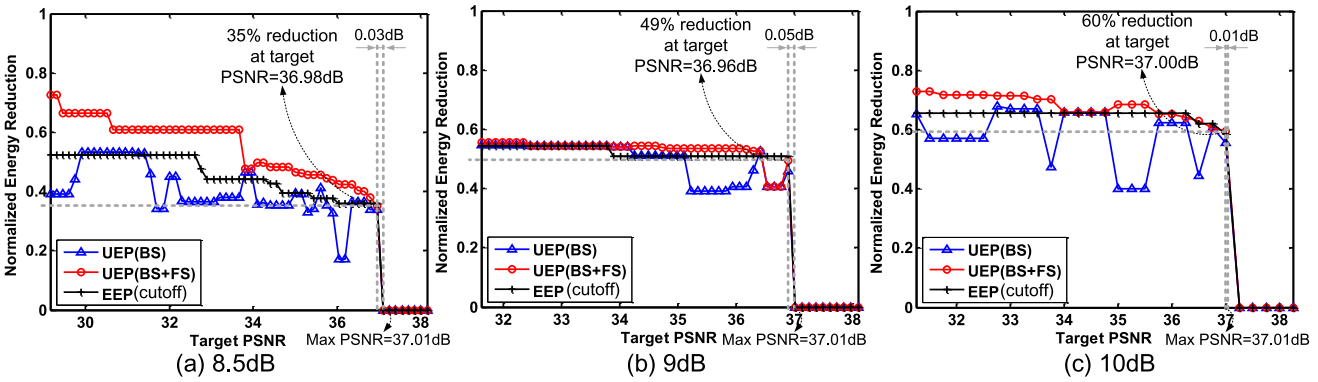

FIGURE 17. Energy reduction results for News: at (a) 8.5 dB, (b) 9 dB, and (c) 10 dB.

C. DISCUSSION

In the preprocess, iteration sets for each priority partition were searched in a target video stream. The searched sets were stored into UEP memory as a lookup table composed of a 1 kbyte SRAM and used in runtime process to optimize power and performance for the joint decoding system. In the runtime process, the MIMO detector received encoded video streams with partition information and generated a read address to read iteration sets from the UEP memory based on a static scheme that used presearched outer and inner iteration sets. In the future work, we may further study to make the preprocess run at runtime when application or channel condition changes.

VI. CONCLUSION

This paper explores the design of a low-power MIMO-LDPC-H.264 joint decoding system using UEP for energy-aware-communication systems. The system is designed to provide an optimal trade-off between energy consumption and performance. Our experimental results demonstrate that the iterative MIMO-LDPC-H.264 joint decoder significantly reduced the overall energy consumption with a negligible reduction in quality. The proposed design achieved 45%, 52%, and 66% energy reduction at 8.5 dB, 9 dB, and 10 dB SNRs, respectively, for Akiyo with 0.08 - 0.09 dB degradations in PSNR when compared to the non-UEP baseline joint source-channel decoder.

REFERENCES

[1] H. Luo, S. Ci, D. Wu, J. Wu, and H. Tang, "Quality-driven cross-layer optimized video delivery over LTE," *IEEE Commun. Mag.*, vol. 48, no. 2, pp. 102–109, Feb. 2010.

[2] A. A. Khalek, C. Caramanis, and R. W. Heath, Jr., "A cross-layer design for perceptual optimization of H.264/SVC with unequal error protection," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 7, pp. 1157–1171, Aug. 2012.

[3] P. Moberg, A. Osseiran, and P. Skillermark, "Cost comparison between SISO and MIMO deployments in future wide area cellular networks," in *Proc. IEEE 69th Veh. Technol. Conf. (VTC Spring)*, Apr. 2009, pp. 1–5.

[4] Y. Shi, W.-H. Kuo, C.-W. Huang, Y.-C. Chou, S.-H. Fang, and D.-N. Yang, "Cross-layer allocation scheme for multi-view videos in massive MIMO networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[5] S.-M. Tseng and Y.-F. Chen, "Average PSNR optimized cross layer user grouping and resource allocation for uplink MU-MIMO OFDMA video communications," *IEEE Access*, vol. 6, pp. 50559–50571, 2018.

[6] S.-J. Kim, G.-Y. Suk, J.-S. Lee, and C.-B. Chae, "QoE-aware scalable video transmission in MIMO systems," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 196–203, Aug. 2017.

[7] D. Bethanabhotla, G. Caire, and M. J. Neely, "WiFlix: Adaptive video streaming in massive MU-MIMO wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 4088–4103, Jun. 2016.

[8] J. Zheng and B. D. Rao, "LDPC-coded MIMO systems with unknown block fading channels: Soft MIMO detector design, channel estimation, and code optimization," *IEEE Trans. Signal Process.*, vol. 54, no. 4, pp. 1504–1518, Apr. 2006.

[9] *International Standard of Joint Video Specification*, document ITU-T Rec. H.264, ISO/IEC 14 496-10 AVC, JVT, 2003.

[10] S. Kumar, L. Xu, M. K. Mandal, and S. Panchanathan, "Error resiliency schemes in H.264/AVC standard," *J. Vis. Commun. Image Represent.*, vol. 17, no. 2, pp. 425–450, 2006.

[11] E. Maani and A. Katsaggelos, "Unequal error protection for robust streaming of scalable video over packet lossy networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 3, pp. 407–416, Mar. 2010.

[12] *T-REC-H.265-High Efficiency Video Coding*, ITU-T, Geneva, Switzerland. Accessed: Feb. 20, 2021.

[13] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[14] N. Thomos, S. Argyropoulos, N. Boulgouris, and M. Strintzis, "Robust transmission of H.264/AVC video using adaptive slice grouping and unequal error protection," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2006, pp. 593–596.

[15] T. Stockhammer and M. Bystrom, "H.264/AVC data partitioning for mobile video communication," in *Proc. Int. Conf. Image Process. (ICIP)*, 2004, pp. 545–548.

[16] W. Heinzelman, M. Budagavi, and R. Talluri, "Unequal error protection of MPEG-4 compressed video," in *Proc. Int. Conf. Image Process. (ICIP)*, vol. 2, 1999, pp. 530–534.

[17] Y. S. Yang, P. Bhagawat, and G. Choi, "Energy-efficient MIMO detection using unequal error protection for embedded joint decoding system," in *Proc. 48th ACM/EDAC/IEEE Design Automat. Conf. (DAC)*, Jun. 2011, pp. 579–584.

[18] Y. S. Yang and G. Choi, "Design space exploration for low-power channel decoder in embedded LDPC-H.264 joint decoding architecture," *Int. J. Inf. Tech. Commun. Converg.*, vol. 1, no. 4, pp. 372–390, 2011.

[19] Y. S. Yang and G. Choi, "Unequal error protection based on DVFS for JSCD in low-power portable multimedia systems," *ACM Trans. Embedded Comput. Syst.*, vol. 11, no. 2, pp. 30:1–30:21, Jul. 2012.

[20] E. Rohani, J. W. Xu, G. Choi, and M. Lu, "Low-power on-the-fly reconfigurable iterative MIMO detection and LDPC decoding design," *Appl. Mech. Mater.*, vols. 496–500, pp. 1825–1829, Jan. 2014.

[21] Y. Huang, M. Ji, J. Sun, B. Wei, and X. Ma, "An unequal coding scheme for H.265 video transmission," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, May 2020, pp. 1–6.

[22] B. Paudel and S. Vafi, "Efficient unequal error protection techniques for tile-based transmission of HEVC videos," *IEEE Access*, vol. 8, pp. 128591–128601, 2020.

[23] B. Zhang, P. Cosman, and L. B. Milstein, "Energy optimization for wireless video transmission employing hybrid ARQ," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5606–5617, Jun. 2019.
[24] M. A. Hosany and G. Ramanna, "Design and implementation of a novel unequal error protection scheme for coded MIMO systems," *AEU-Int. J. Electron. Commun.*, vol. 122, Jul. 2020, Art. no. 153257.
[25] J. Park, Y. Yang, and J. Kim, "Unequal error protection technique for video streaming over MIMO-OFDM systems," *PLoS ONE*, vol. 14, pp. 1–18, Jan. 2019.
[26] Y. Zhang, C. Xu, I. A. Hemadeh, M. El-Hajjar, and L. Hanzo, "Near-instantaneously adaptive multi-set space-time shift keying for UAV-aided video surveillance," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12843–12856, Nov. 2020.
[27] G.-H. Yang, D. Shen, and V. O. K. Li, "Unequal error protection for MIMO systems with a hybrid structure," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, pp. 684–685.
[28] Y. Liu, Q. S. Tong, A. D. Men, Z. Y. Quan, and B. Yang, "A joint source-channel coding scheme focused on unequal error protection for H.264 transmission over MIMO-OFDM system," in *Proc. Int. Colloq. Comput., Commun., Control, Manage.*, vol. 2, 2008, pp. 491–495.
[29] X. Li, S. Yang, Y. Wang, and Z. Li, "Performance of UEP based MIMO scheme for LDPC codes," in *Proc. Int. Conf. Comput. Eng. Technol. (ICCET)*, vol. 6, 2010, pp. 216–220.
[30] R. Guo, "Stereo video transmission using LDPC code," *Int. J. Commun., Netw. Syst. Sci.*, vol. 1, no. 3, pp. 254–259, 2008.
[31] Y. Wang, S. Yu, and X. Yang, "Error robustness scheme for H.264 based on LDPC code," in *Proc. Int. Multi-Media Modelling Conf.*, 2006, p. 4.
[32] L. Qi, L. Yang, W. Wensheng, C. Huijuan, and T. Kun, "Robust video transmission scheme using dynamic rate selection LDPC and RS codes," in *Proc. IMACS Multiconf. Comput. Eng. Syst. Appl.*, vol. 2, 2006, pp. 1673–1679.
[33] Y. Wang, S. Yu, and X. Yang, "Unequal iterative decoding for power efficient video transmission," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2006, pp. 613–616.
[34] X. Lu, E. Erkip, Y. Wang, and D. Goodman, "Power efficient multimedia communication over wireless channels," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 10, pp. 1738–1751, Dec. 2003.
[35] Q. Zhang, Z. Ji, W. Zhu, and Y.-Q. Zhang, "Power-minimized bit allocation for video communication over wireless channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 398–410, Jun. 2002.
[36] Y. Eisenberg, C. E. Luna, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Joint source coding and transmission power management for energy efficient wireless video communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 411–424, Jun. 2002.
[37] J. Dielissen, A. Hekstra, and V. Berg, "Low cost LDPC decoder for DVB-S2," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, vol. 2, 2006, pp. 1–6.
[38] Z. Ma and S. Sun, "Research on HEVC screen content coding and video transmission technology based on machine learning," *Ad Hoc Netw.*, vol. 107, Oct. 2020, Art. no. 102257.
[39] R. Hoefel, "IEEE 802.11ax: On performance of multi-antenna technologies with LDPC codes," in *Proc. IEEE 7th Int. Conf. Commun. Electron. (ICCE)*, Jul. 2018, pp. 159–164.
[40] W. Tang, C.-H. Chen, and Z. Zhang, "A 2.4-mm^2 130-mW MMSE-nonbinary LDPC iterative detector decoder for 4 × 4 256-QAM MIMO in 65-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 54, no. 7, pp. 2070–2080, Jul. 2019.
[41] A. G. D. Uchoa, C. T. Healy, and R. C. D. Lamare, "Iterative detection and decoding algorithms for MIMO systems in block-fading channels using LDPC codes," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2735–2741, Apr. 2016.
[42] K. Wang and Z. Ding, "Joint turbo receiver for LDPC-coded MIMO systems based on semi-definite relaxation," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–5.
[43] A. Ivanov, A. Savinov, and D. Yarotsky, "Iterative nonlinear detection and decoding in multi-user massive MIMO," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 573–578.
[44] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.
[45] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 33, no. 6, pp. 457–458, 1997.
[46] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 5, pp. 533–547, Sep. 1981.
[47] F. Kienle and N. Wehn, "Low complexity stopping criterion for LDPC code decoders," in *Proc. Veh. Technol. Conf. (VTC)*, vol. 1, 2005, pp. 606–609.
[48] G. Glikiotis and V. Paliouras, "A low-power termination criterion for iterative LDPC code decoders," in *Proc. IEEE Workshop Signal Process. Syst. Design Implement.*, 2005, pp. 122–127.
[49] W. Wang and G. Choi, "Minimum-energy LDPC decoder for real-time mobile application," in *Proc. Design, Automat. Test Eur. Conf. Exhib.*, Apr. 2007, pp. 1–6.
[50] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389–399, Mar. 2003.
[51] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bolcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, Jul. 2005.
[52] *ITU-T H.264: Advanced Video Coding for Generic Audiovisual Services*, document H.264/AVC JVT JM-18.3, VCEG. Accessed: Feb. 20, 2021.
[53] H. Kim, D. Lee, and J. Villasenor, "Design tradeoffs and hardware architecture for real-time iterative MIMO detection using sphere decoding and LDPC coding," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 6, pp. 1003–1014, Aug. 2008.
[54] M. Myllyla, J. Antikainen, M. Juntti, and J. R. Cavallaro, "The effect of LLR clipping to the complexity of list sphere detector algorithms," in *Proc. 41st Asilomar Conf. Signals, Syst. Comput. (ACSSC)*, Nov. 2007, pp. 1559–1563.
[55] K. K. Gunnam, G. S. Choi, W. Wang, E. Kim, and M. B. Yeary, "Decoding of quasi-cyclic LDPC codes using an on-the-fly computation," in *Proc. 40th Asilomar Conf. Signals, Syst. Comput. (ACSSC)*, 2006, pp. 1192–1199.
[56] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.

YOON SEOK YANG (Member, IEEE) received the B.S. and M.S. degrees from Hanyang University, Seoul, South Korea, in 1998 and 2000, respectively, the M.S. degree in electrical engineering and computer science from the University of California at Irvine, Irvine, CA, USA, in 2008, and the Ph.D. degree in electrical and computer engineering from Texas A&M University, College Station, TX, USA, in 2012. From 2000 to 2005, he was a Research Engineer with the Digital TV Laboratory, LG Electronics, Seoul. He is currently a Research Scientist with Intel Corporation, Santa Clara, CA, USA. His current research interests include neuromorphic computing, system-on-chip (SoC) design for artificial intelligence (AI), network-on-chip (NoC) design for multiprocessor architecture, signal processing for machine learning, and edge computing.

YONGTAE KIM (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Korea University, Seoul, Republic of Korea, in 2007 and 2009, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA, in 2013. From 2013 to 2018, he was a Software Engineer with Intel Corporation, Santa Clara, CA, USA. Since 2018, he has been with the School of Computer Science and Engineering, Kyungpook National University, Daegu, Republic of Korea, where he is currently an Assistant Professor. His research interests include energy efficient integrated circuits and systems, particularly, neuromorphic computing and approximate computing, and new memory devices and architectures.