This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2527303, IEEE Transactions on Circuits and Systems for Video Technology

Error Resilient Video Encoding Using Parallel Independent Signature Processing

Joshua W. Wells, *Member, IEEE,* and Abhijit Chatterjee, *Fellow, IEEE*

***Abstract*—Soft errors resulting from encoding video sequences on unreliable hardware can create significant artifacts in decoded video sequences, contributing to extreme video quality degradation. Modern systems are required to operate under increasingly challenging constraints including smaller feature sizes and lower operating voltage, increasing the likelihood of soft errors in the video encoding hardware. These conditions are of particular concern for energy limited, battery operated systems since they may be required to operate in non-ideal environments and/or continue operating with a practically depleted energy source. The proposed *parallel independent signature processing* design performs error detection and mitigation in video encoding hardware, enabling a graceful degradation of quality when encoding using unreliable hardware. The effects of soft errors are minimized by preventing the error propagation normally associated with errors in encoded video sequences. This allows for the recovery of quality when errors are present in the video encoding system. Conventional video encoding techniques are designed to handle worst-case error rates by increasing gate sizes and/or increasing the operating voltage of the system. Such designs have error-rate limits and when these limits are reached, the systems tend to fail catastrophically resulting in an unrecoverable signal. The proposed design allows for single upset events to translate to single, transient artifacts in a decoded video sequence.**

***Index Terms*—video coding, encoding, video signal processing, adaptive signal processing, error analysis, error correction**

I. Introduction

The goal of video encoding is to eliminate redundant information within a video signal, keeping only the information necessary to perform an acceptable reconstruction of the signal, resulting in compression. A necessary side effect of compression is increased error sensitivity. A single bit error in an uncompressed video signal will have limited effects, only changing the appearance of a single pixel. But a similar, single bit error in an encoded video signal can potentially induce errors in a very large number pixels of the decoded video signal due to data interdependence. Traditional efforts to protect encoded video signals focus on protecting the channel between the video encoder and the decoder—formerly the most probable source of error—while the encoder and decoder hardware are designed for worst case scenarios. However, continued device scaling and lower operating voltage are making reliable hardware design an increasingly difficult challenge [1], [2].

J. W. Wells and A. Chatterjee are with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA e-mail: josh.wells@gatech.edu

This material is based upon work supported by the National Science Foundation under Grant No. CCR-0834484.

Soft errors [3] in either the encoding or decoding hardware can have a significant impact on the quality of the decoded signal as the errors can have a direct effect on the encoded signal. The design of most modern coder-decoders (CODECs) place a much larger computational load on the video encoder compared to the decoder, creating a need for faster processing at the encoder. Demand for higher resolution and real-time encoding further increases processing requirements for video encoding hardware. This requires increased technology scaling in the video encoder (but not necessarily for the decoder). This demand for continued technology scaling creates an increasing probability of soft errors in video encoding hardware [4]. Additionally, errors can occur because of reduced noise margins and excessive path delays incurred by low voltage operation of the encoder, causing periodic upsets in the logic values latched into the circuit flip flops [5], [6], [7], [8]. The increasing likelihood of errors in video encoding hardware necessitating design methods unreliable hardware

Prior work has concentrated on protecting either the encoded video signal or specific subsystems of the encoder hardware. The encoder, as a whole, is still susceptible to soft errors and thus, susceptible to extreme degradation of encoded signal quality. The research presented provides a design method for motion-compensated hybrid video encoders (MCHVEs) capable of controlling the effects of soft errors in the video encoder by identifying sources of propagating error and altering encoding modes to prevent errors from propagating, making the visual artifacts of errors in the encoded signal transient. All subsystems of the video encoder are accounted for with the exception of the context-adaptive variable-length coding (CAVLC) system. This system is outside the feedback loop created by the video encoder and can be protected using methods described in [9], [10]. This effectively allows for the a graceful degradation of encoded video signal quality with increasing soft error rates.

Throughout this paper, vector notation is used to represent macroblock (MB) variables because the systems interacting with the MBs are modeled as linear transforms (when possible). Variables representing vectors are indicated with bold, upright, lowercase letters. When necessary, MBs may also be referenced using 2-D matrix notation indicated by an uppercase, bold variable. Additionally, video sequences are assumed to be sampled using *YV12*. MB data is converted from a three plane matrix form to a vector by raster ordering the samples for each color plane.

The remainder of this paper is organized as follows. Prior work in the realm of error resilient video encoding is reviewed in Section II. An overview of motion-compensated hybrid

CODECs is provided in Section III to provide a basis for the proposed work. The proposed parallel independent signature processing (PISP) design is discussed in Section IV. Experimental results are presented and discussed in Section V. Finally, conclusions are drawn in Section VI and future applications of PISP are hypothesized.

II. Prior Work in Error Resilient Video Encoding

Part of the work presented in the paper was previously published in [11], though, the majority of the content is novel. The general concept of using checksums to protect MB information and the proposed error mitigation method exist in the previous publication. All other content presented in this paper including *a*) the vulnerable system analysis, *b*) the signature generation method, *c*) parallel independent signature processing methodology, and *d*) detailed results are wholly novel.

Prior work in error resilient video encoding and decoding can be grouped into two primary categories based on the source of the errors being protected against: *a*) the channel between the video encoder and decoder, and *b*) the video encoding hardware for certain subsystems. The video decoding hardware can more easily be designed to be reliable and is not considered an important source of computational error.

A. Channel Protection

Channel protection for encoded video data uses specific techniques for mitigating channel errors that go beyond standard forward error correction (FEC). In [12], a combination of two effective priority-based FEC methods are used to produce a higher quality signal than either method alone. In addition to adding error resilience to the encoded signal, [13] evaluates the implied impact on power and offers an optimized solution. These methods are implemented solely in the encoding hardware—the level of protection required for the channel must be assumed by the encoder. Other research has attempted to improve these assumptions by proposing the addition of a feedback channel to communicate the condition of the channel back to the encoder or communicating node [14], [15], [16], [17]. These methods perform error detection at the decoder and communicate the state of the decoder back to the encoder using a feedback channel. This allows for the amount of redundancy in the video signal to be dynamically tuned to provide minimal protection to correctly decode the transmitted signal. Various error resilient techniques have also been incorporated into video coding standards. The ISO MPEG-4 standard allowed for video packet resynchronization, data partitioning, reversible variable-length codes (VLCs), and header extension code [18]. These methods attempt to confine and minimize the effects of the channel errors and are available in most modern standards [19].

B. Encoding Hardware Protection

Prior work in mitigating errors induced by the video encoding hardware has primarily concentrated on two subsystems of the video encoder: the motion estimation (ME) system and the VLC system.

Most of the processing performed by a MCHVE is typically performed by the ME system. As a result, the problem of ensuring error-free computation in this system has received concentrated attention. In [20], [21], an error-tolerant ME algorithm is used with potentially unreliable hardware to increase the likelihood that good—though possibly not precise—motion vectors are selected. Other work has performed a hardware design analysis of the ME system in an effort to identify devices that should be selectively hardened to protect against soft errors [22]. Other work has intentionally allowed errors to occur in the ME system by over-scaling the voltage [23], [24] supply. Low resolution estimates (which are computed correctly with the over-scaled voltage) are used to detect computational errors in the ME system. The estimates are used in place of incorrectly computed results when an error is detected. Although an error in the ME system may produce a non-optimal motion vector, the error may not impact the quality of the decoded signal. Instead, ME errors are more likely to affect the compression performance of the video encoder.

Prior work in error resilient VLC systems has focused specifically on CAVLC. CAVLC uses a varying VLC code based on previously coded parts of the video signal, allowing for additional compression. To provide error detection for the CAVLC system, a correlation between selected quantization parameter (QP) and VLC table selection can be utilized [10], [9]. Error correction is performed by comparing potentially incorrect values with redundant values.

Prior works targeting specific systems of video encoders are successful in their defined scope, but they do not effectively address the issue of making the video encoder, as a whole, error resilient. When a video encoder is subjected to an increasingly large number of errors, the effects of the errors are not likely to be confined to a single system. The proposed design aims to protect the entire video encoding process, up to the VLC system.

III. Video Encoding Overview

The class of video encoders considered in this paper is the motion-compensated hybrid video encoder (MCHVE). There are many standards that fall into this class of CODEC, but H.264 [25] will be used as a reference for experiments, due to the standard being arguably one of the most widely accepted and implemented standards at present. MCHVEs process each frame (picture), sequentially, by dividing the frame into an integer number of non-overlapping, contiguous blocks called macroblocks (MBs). Each MB contains the same number of samples (pixels) and each MB is encoded individually (but not necessarily independently). The MBs, are processed in raster order within each frame. The three primary stages of the video encoder, *a*) prediction, *b*) transform, and *c*) reconstruction, are shown in Fig. 1a. Each sampled MB, $\mathbf{b}_s[n]$, is processed by the prediction system, followed by the transform system to produce the encoded MB, $\mathbf{b}_{enc}[n]$. Each encoded MB is transmitted to the decoder(s) along with instructions ($\mathbf{v}$ or a from Fig. 1a) indicating modes of operation for decoding. Each sampled MB is described in terms of a previously encoded

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2527303, IEEE Transactions on Circuits and Systems for Video Technology

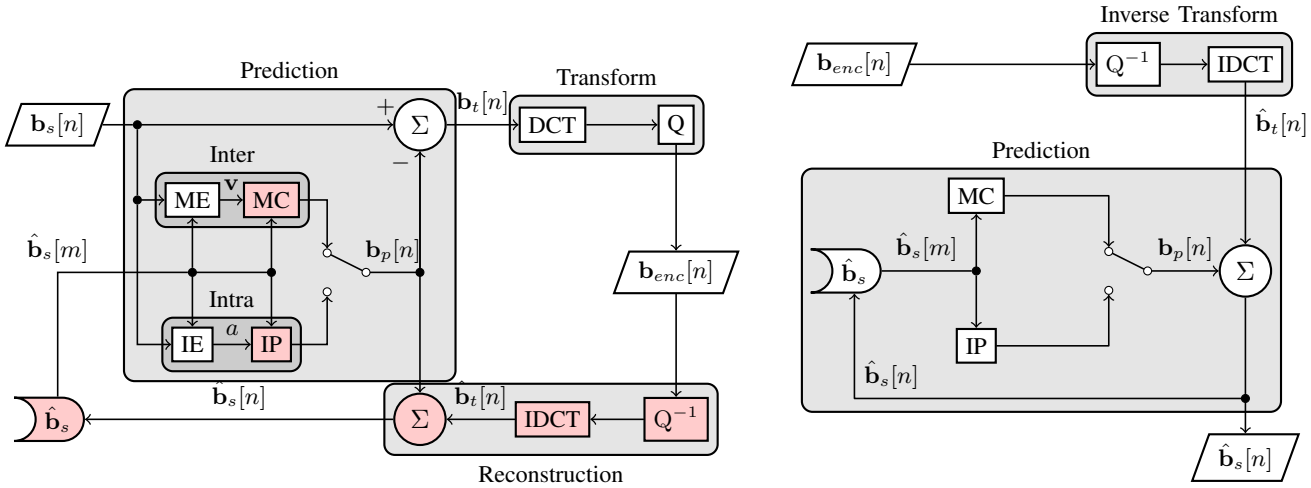

Fig. 1. Conventional system design of a MCHVE (a) and MCHVD (b) with sampled MB input $\mathbf{b}_s$, encoded MB output $\mathbf{b}_{enc}$, and decoded MB reference $\hat{\mathbf{b}}_s$. Shaded encoder subsystems mark vulnerabilities to decoder drift.

MB, $\hat{\mathbf{b}}_s[m]$. The index used for each MB is associated with the order in which the MBs are processed. In Fig. 1, the current MB being encoded is indexed by n, and m represents the index of a previously encoded MB. Therefore, $m < n$.

The MB decoding process, shown in 1b, performs an inverse transform followed by the adding of a prediction of type signaled by the encoder. The result is the reconstructed MB, $\hat{\mathbf{b}}_s[n]$, which will differ slightly from the original MB, $\mathbf{b}_s[n]$, due to quantization noise introduced by the quantization operation. The prediction system, transform system, and reconstruction system are discussed in more detail in the following subsections.

A. Prediction

The prediction system functions by evaluating a number of prediction modes and selecting the mode that produces the greatest compression ratio (or another desirable metric) [26]. There are two primary types of modes for prediction: intra prediction and inter prediction. As indicated in Fig. 1a, either intra prediction or inter prediction can be selected and intra estimation (IE) or ME will be used to select the best performing mode. In either case, previously encoded MBs are used as a basis for describing the MB currently being encoded. Inter prediction uses MBs encoded in previous frames while intra prediction uses previously encoded MBs in the current frame. After a prediction method has been selected, the selection is signaled via a (intra prediction mode) or $\mathbf{v}$ (inter prediction motion vector) to the motion compensation (MC) or intra prediction (IP) system depending on which prediction mode is being used. Either the MC or IP system will create the prediction, $\mathbf{b}_p$, which will be subtracted from the sampled MB to produce the residual MB, $\mathbf{b}_t$. This same prediction will be added back after the residual MB has been transformed and inverse transformed.

B. Transform

The transform system reduces the amount of redundant information within each residual MB by transforming each 4×4 block within each MB using a 2-D discrete cosine transform (DCT), followed by quantization (Q) as shown in Fig. 1a. The resulting non-zero coefficients are finally encoded using an entropy encoder, which has been omitted for simplicity. When combined with the prediction mode ($\mathbf{v}$ or a), the decoder has all necessary information to reconstruct each MB.

After transforming and quantization, each MB is inverse transformed with a 2-D inverse discrete cosine transform (IDCT) to produce $\hat{\mathbf{b}}_t$, a reconstruction of $\mathbf{b}_t$. Due to quantization and transform coefficient rounding, some quantization error is added to the reconstructed signal, $\hat{\mathbf{b}}_t$. The prediction is finally added to the reconstructed, residual MB to produce the final reconstructed MB, $\hat{\mathbf{b}}_s$.

C. Reconstruction and Decoder Drift

Essential to proper operation of the encoder and decoder is the state of the reference MBs, $\hat{\mathbf{b}}_s$. If the decoder memory differs from the encoder memory, an effect called *decoder drift* occurs. For this reason, the exact MB reconstruction performed on the decoder is also performed in the encoder (along with quantization error), as indicated by the shaded systems in Fig. 1a. If decoder drift occurs, artifacts caused by memory differences can propagate from one MB to another. If an erroneous MB is used to predict other MBs, the resulting, encoded MBs may also be in error. Thus, errors can be propagated from one MB to others.

Properly functioning encoders and decoders are designed to prevent decoder drift. However, soft errors may allow a decoder drift to occur. Specifically, errors occurring in any of the shaded systems in Fig. 1a can cause a drift, allowing errors to propagate. If an operation by a shaded system is performed differently in the encoder than its partner system in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2527303, IEEE Transactions on Circuits and Systems for Video Technology

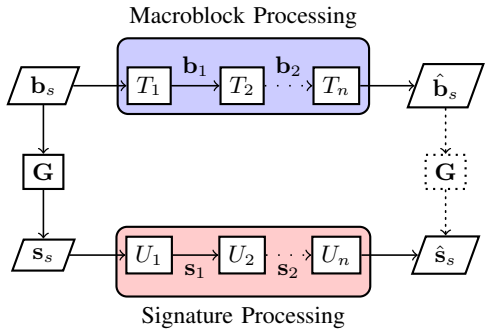

Fig. 2. Ideal PISP theoretical overview with linear systems. Signature processing produces a signature matching the output of MB processing.

the decoder, the values recorded in memories of the encoder and decoder for an MB will be different. Any future prediction made from this incoherent MB will create more unmatched MBs. In this way, errors are propagated and the memory of the decoder drifts away from the memory of the encoder. It is, therefore, imperative to develop protection for all vulnerable systems of the video encoder, especially when using unreliable hardware. Section IV describes the proposed method for protecting the vulnerable encoding systems through error detection and mitigation.

IV. Parallel Independent Signature Processing

The novel contribution of this paper is a video encoder design methodology referred to as parallel independent signature processing (PISP). PISP is a method of performing error detection and mitigation of error effects by creating a compact signature for each sampled MB, and processing each signature in parallel with the MB it was created from. Fig. 2 demonstrates this principle by generalizing a video encoder as a series of systems (T_i) operating on the sampled MB, $\mathbf{b}_s$, to yield the reconstructed MB, $\hat{\mathbf{b}}_s$. The systems shown in Fig. 2 can represent any system or collection of systems from Fig. 1a. For instance, T_1 might represent the prediction system; T_2 might represent the transform system (or perhaps one of its subsystems); and finally, the reconstructed MB, $\hat{\mathbf{b}}_s$ is produced. The signature for the sampled MB, $\mathbf{s}_s$, is generated by the signature generating matrix, $\mathbf{G}$. As the MB is processed by the MB processing systems, T_i, the signature is processed by analogous systems, U_i. The practicality of the PISP design is subject to the selection of systems, U_i, such that the following equation holds:

$$\mathbf{G}\mathbf{b}_i = \mathbf{s}_i, \forall i \tag{1}$$

If the necessary signature processing systems exist under the constraint in (1), an incorrectly computed result for any of the MB processing systems will produce a final reconstructed MB that does not match the corresponding, reconstructed signature. Errors can, therefore, be detected by comparing $\mathbf{G}\hat{\mathbf{b}}_s$ to $\hat{\mathbf{s}}_s$. If the two results match, the MB and signature were processed correctly with high probability. Otherwise, an incorrect result was produced by one of the MB or signature processing systems.

In general, the length of the signature for a given MB is expected to be much smaller than the length of the MB vector. Therefore, the generating matrix, $\mathbf{G}$, reduces dimensionality, making the generalized design in Fig. 2 impractical. To compensate for this, a differential signature system, $\mathbf{D}$, is introduced in Fig. 3. A particular differential signature system, $\mathbf{D}_i$, computes a differential signature vector, $\mathbf{s}_{d_i}$, that represents the difference between $\mathbf{s}_i$ and $\mathbf{s}_{i-1}$. The computation is performed as a function of the input to the corresponding MB processing system, $\mathbf{T}_i$. The differential signature is defined as

$$\begin{aligned}\mathbf{s}_{d_i} &= \mathbf{s}_i - \mathbf{s}_{i-1} \\ &= \mathbf{G}(\mathbf{b}_i - \mathbf{b}_{i-1}) \pmod{2^l},\end{aligned} \tag{2}$$

however, the computation must be performed a priori, relative to the MB processing system, T_i—the output of the system cannot be observed directly. To enable this, each MB processing system is modeled as linear transform (when possible), which is indicated by the notation change in Fig. 3. This assumption allows (2) to be redefined as

$$\begin{aligned}\mathbf{s}_{d_i} &= \mathbf{G}\mathbf{b}_i - \mathbf{G}\mathbf{b}_{i-1} \\ &= \mathbf{G}\mathbf{T}_i\mathbf{b}_{i-1} - \mathbf{G}\mathbf{b}_{i-1} \\ &= \mathbf{G}(\mathbf{T}_i - \mathbf{I})\mathbf{b}_{i-1} \\ &= \mathbf{D}_i\mathbf{b}_{i-1} \pmod{2^l},\end{aligned} \tag{3}$$

where l is the bit depth of the the pixels samples, and the differential signature systems are defined as

$$\mathbf{D}_i = \mathbf{G}(\mathbf{T}_i - \mathbf{I}). \tag{4}$$

As indicated by Fig. 3, the previously defined signature processing systems, U_i, are reduced to a summation with a differential signature. The output of each system is defined as,

$$\begin{aligned}\mathbf{s}_i &= \mathbf{s}_{i-1} + \mathbf{s}_{d_i} \\ &= \mathbf{s}_{i-1} + \mathbf{D}_i\mathbf{b}_{i-1} \pmod{2^l}.\end{aligned} \tag{5}$$

If each MB processing system can be modeled as a linear transform, independent signature processing can be realized as shown in Fig. 3. Furthermore, once a generating matrix has been selected, each differential signature system matrix can be precomputed once, reducing the amount of overhead added by the proposed design.

Not all systems of a video encoder are linear. These systems cannot be incorporated into the proposed method in Fig. 3 directly. Instead, the nonlinear operations in these systems (such as rounding in the quantization system) are modeled using additive noise which can be directly measured and used to adjust differential signatures. The use of this method is detailed further in Section IV-C.

The signature generating matrix, $\mathbf{G}$, can represent any linear block code (LBC). The generated code word functions as a signature for the data, so the length of the code should be much smaller than the length of a MB vector. Code parameters can be changed to provide more or less robust error detection capability. To minimize computational overhead, the generating matrix needs to be sparse. Additionally, modular arithmetic is used to maintain a tight signature space and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2527303, IEEE Transactions on Circuits and Systems for Video Technology

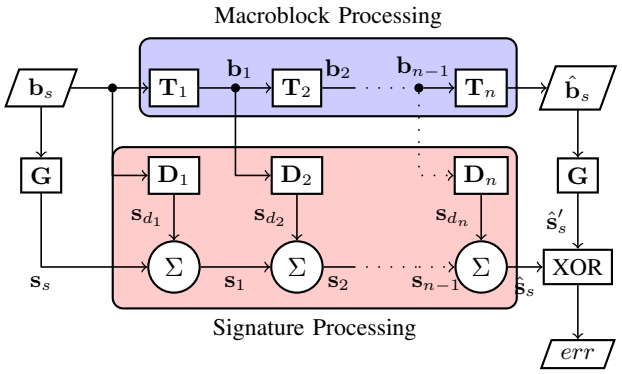

Fig. 3. Realizable PISP Overview. Differential systems, $\mathbf{D}_i$, are used to provide missing information to signature processing system.

reduce computational overhead for signatures. The modulo is 2^l, where l is the bit depth of the MB samples.

Although there are many generating matrices that will satisfy the design constraints, for practical reasons, a pseudo-random, sparse, binary generating matrix is used with the design. Each element of the matrix is assigned a value of 1 or 0 with some preference to 0 while ensuring each column of the matrix contains at least one non-zero value. The use of binary values avoids expensive multiply operations when computing signatures and when computing differential signatures (explained later in this section).

MBs are typically represented as multiple matrices, each representing a single color plane. However, in this paper, each matrix is converted to a raster ordered vector of the matrix elements. Each color plane is appended in order. The H.264 standard may use different prediction modes for different planes. As a result, the matrix representing the linear transform for an MB processing system is generally sparse. A general MB processing system modeled as a linear transform will take the form

$$\mathbf{T}_i = \begin{bmatrix} \mathbf{T}_{i_Y} & 0 & 0 \\ 0 & \mathbf{T}_{i_{Cb}} & 0 \\ 0 & 0 & \mathbf{T}_{i_{Cr}} \end{bmatrix}, \tag{6}$$

where each of the matrices along the diagonal performs a linear transform on a single color plane indicated by subscript (luma, blue chrominance, and red chrominance). Throughout this paper, when a transform is described, only the primary color plane will be addressed to avoid being overly verbose. The same transform can easily be scaled and applied to other color planes.

The application of the generalized PISP design to a conventional MCHVE yields the design shown in Fig. 4. The MB processing systems and signature processing systems are analogous to the generalized counterparts in Fig. 2. The development of the necessary differential systems for the specific design is discussed in later sections. The prediction and transform systems contain all processing elements depicted in Fig. 1a. Signature processing is performed in parallel with MB processing. As each MB is encoded, reconstructed, and stored in memory, a corresponding signature is generated, processed, and stored. Any time a stored MB

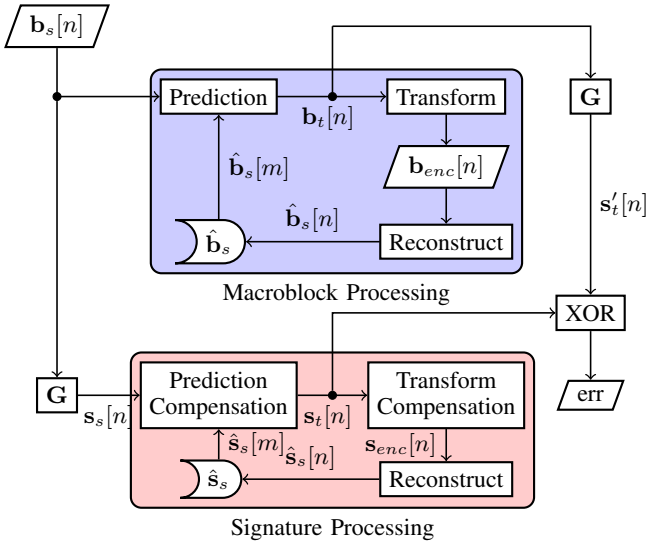

Fig. 4. Proposed video encoder design

is used as a prediction reference to produce a residual MB, $\mathbf{b}_t[n]$, the corresponding, stored signature will be used to generate the prediction signature, $\mathbf{s}_t[n]$. For each processed MB, the residual signature is checked for coherence with the associated, residual MB. Performing the comparison after prediction allows for validation of proper processing by the prediction system. Performing validation at any other point may not detect errors occurring within the prediction systems (MC and IP) as the same, potentially erroneous, prediction is added back during reconstruction (shown in Fig. 1a), obscuring any potential errors that may exist in an encoded MB. The validation signature generated directly from the residual MB, $\mathbf{s}'_t[n]$, and the residual signature, $\mathbf{s}_t[n]$, are compared. If the signatures match, correct processing up to this point is highly likely. Otherwise, non-matching signatures indicate that a processing error has occurred and mitigating action should be initiated.

In the following subsections, the specifics of the proposed encoder design are presented. Section IV-A validates the ability of the proposed design to detect errors. Section IV-B analyzes the prediction system of the conventional encoder design and develops the necessary prediction signature processing system of the proposed encoder design. Section IV-C analyzes both the transform and decoding systems of the conventional encoder design and develops the transform signature processing system of the proposed encoder design. Finally, error detection and mitigation methods are presented in Section IV-D.

A. PISP Design Validation

General validation of the proposed method is performed by evaluating the effects of additive error from the system, $\mathbf{T}_{i-1}$, in Fig. 3. For the design to be practical, a validation signature generated directly from a particular MB must be incoherent

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2527303, IEEE Transactions on Circuits and Systems for Video Technology

with the corresponding signature calculated by the signature processing system when the MB vector is in error:

$$\mathbf{G}\mathbf{b}_i \neq \mathbf{s}_i \quad \text{if } \mathbf{b}_i \text{ is erroneous.} \tag{7}$$

Assuming an error is made by system $\mathbf{T}_{i-1}$, let the erroneous output from the system be defined as

$$\tilde{\mathbf{b}}_{i-1} = \mathbf{b}_{i-1} + \mathbf{e}, \tag{8}$$

where $\mathbf{e}$ is additive error and $\mathbf{b}_{i-1}$ is the correct result. The next MB processing system, $\mathbf{T}_i$, uses the incorrect result producing,

$$\begin{aligned}\tilde{\mathbf{b}}_i &= \mathbf{T}_i\tilde{\mathbf{b}}_{i-1} \\ &= \mathbf{T}_i(\mathbf{b}_{i-1} + \mathbf{e}).\end{aligned} \tag{9}$$

The corresponding signature computed using the erroneous MB is

$$\begin{aligned}\tilde{\mathbf{s}}_i &= \mathbf{s}_{i-1} + \mathbf{D}_i\left(\mathbf{b}_{i-1} + \mathbf{e}\right) \\ &= \mathbf{s}_{i-1} + \mathbf{D}_i\mathbf{b}_{i-1} + \mathbf{D}_i\mathbf{e} \\ &= \mathbf{s}_i + \mathbf{D}_i\mathbf{e} \\ &= \mathbf{s}_i + \mathbf{G}\left(\mathbf{T}_i - \mathbf{I}\right)\mathbf{e} \\ &= \mathbf{s}_i + \mathbf{G}\mathbf{T}_i\mathbf{e} - \mathbf{G}\mathbf{e} \pmod{2^l}.\end{aligned} \tag{10}$$

Direct evaluation of the MB result yields the following validation signature:

$$\begin{aligned}\tilde{\mathbf{s}}'_i &= \mathbf{G}\tilde{\mathbf{b}}_i \\ &= \mathbf{G}\mathbf{T}_i\mathbf{b}_{i-1} + \mathbf{G}\mathbf{T}_i\mathbf{e} \\ &= \mathbf{s}_i + \mathbf{G}\mathbf{T}_i\mathbf{e} \pmod{2^l}.\end{aligned} \tag{11}$$

A comparison of (10) and (11) reveals the extra term $-\mathbf{G}\mathbf{e}$ in the former, indicating that although an error in the MB processing path will affect both the MB and signature processing systems, the signature will be incoherent with the MB as long as the error is not in the null space of the signature generating matrix. Therefore, the signature will diverge from its counterpart MB whenever errors are present in the MB signal, with high probability. The divergence can be effectively used for error detection.

B. Prediction Signature Processing

Prediction signature processing is performed by modeling the entire prediction system as two, sequential, linear transforms, emphasized in Fig. 5. The first modeled system consists of the MC or IP system, depending on the type of prediction being used. The second modeled system is the subtraction operation where the predicted MB is subtracted from the sampled one. An analogous signature processing system is shown in Fig. 5 for both MB processing systems. The ME and IE systems analyze the sampled MB to select the appropriate prediction mode and communicate the selected mode to the MC or IP system, respectively. Differing modes of prediction imply differing linear transforms to be used by the MC/IP system. Likewise, the selected mode ($\mathbf{v}$ or a) is communicated to the differential signature system so an appropriate linear transform can be selected. The transforms, therefore, depend on the prediction mode, and the generating matrix, $\mathbf{G}$. Once

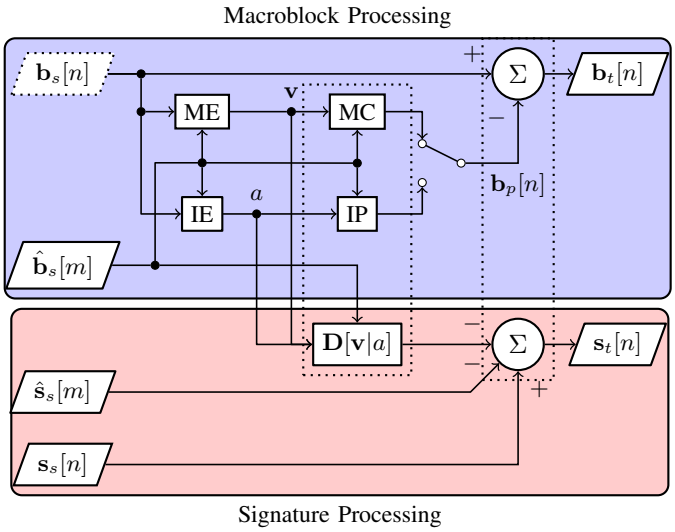

Fig. 5. Prediction Signature Processing Model Overview

$\mathbf{G}$ is selected, each $\mathbf{D}$ can be precomputed and cached to conserve computational effort. Each $\mathbf{D}$ matrix is of the same dimensions as $\mathbf{G}$, therefore, storing a number of the transforms is feasible. The residual MB is defined as

$$\begin{aligned}\mathbf{b}_t[n] &= \mathbf{b}_s[n] - \mathbf{b}_p[n] \\ &= \mathbf{b}_s[n] - \mathbf{T}\hat{\mathbf{b}}_s[m],\end{aligned} \tag{12}$$

where $\mathbf{T}$ is the (linear transform modeled) prediction matrix.

The first modeled system (MC/IP) operates strictly on the previously encoded MB, $\hat{\mathbf{b}}_s[m]$, given the selected prediction mode. The resulting prediction is defined as

$$\mathbf{b}_p[n] = \mathbf{T}\hat{\mathbf{b}}_s[m], \tag{13}$$

where $\mathbf{T}$ is the prediction transform used by either the MC system or the IP system. The value of $\mathbf{T}$ varies depending on the selected prediction mode. The associated, processed signature is defined as

$$\mathbf{s}_p[n] = \hat{\mathbf{s}}_s[m] + \mathbf{D}\hat{\mathbf{b}}_s[m] \pmod{2^l}, \tag{14}$$

where $\hat{\mathbf{s}}_s[m]$ is the signature associated with the stored MB, $\hat{\mathbf{b}}_s[m]$, and $\mathbf{D}$ is the differential signature system associated with the selected MB prediction transform. The derivation of $\mathbf{T}$ and $\mathbf{D}$ are described in the following subsections.

The second modeled system in Fig. 5 is a subtraction operation that operates on the prediction MB and the sampled MB. The resulting residual MB is defined as

$$\mathbf{b}_t[n] = \mathbf{b}_s[n] - \mathbf{b}_p[n]. \tag{15}$$

Because the subtraction operation is a linear operator, a differential signature transform is not needed. Instead, the same linear operator can be applied to the corresponding signatures. The residual signature is defined as

$$\begin{aligned}\mathbf{s}_t[n] &= \mathbf{s}_s[n] - \mathbf{s}_p[n] \\ &= \mathbf{s}_s[n] - \hat{\mathbf{s}}_s[m] - \mathbf{D}\hat{\mathbf{b}}_s[m] \pmod{2^l},\end{aligned} \tag{16}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2527303, IEEE Transactions on Circuits and Systems for Video Technology

which is the implementation shown in Fig. 5.

Although the prediction signature processing definition in (16) works well for many simple prediction modes that base predictions on a single MB, there are also prediction modes that use information from multiple stored MBs to create a single prediction. To accommodate these prediction modes, (12) is redefined as

$$\mathbf{b}_t[n] = \mathbf{b}_s[n] - \sum_m \mathbf{T}[m-n]\hat{\mathbf{b}}_s[m], \tag{17}$$

allowing the residual MB to be described as a sum of multiple predictions. Valid values for m depend on the type of prediction being used and are described in detail in the following subsections. Similarly, (16) is redefined as

$$\mathbf{s}_t[n] = \mathbf{s}_s[n] - \sum_m \hat{\mathbf{s}}_s[m] - \sum_m \mathbf{D}[m-n]\hat{\mathbf{b}}_s[m] \pmod{2^l}. \tag{18}$$

This allows for more complex prediction modes to be used, such as an MC prediction that spans multiple (up to 4) MBs. The details of the prediction transformations for intra and inter prediction are described in detail in Section IV-B1 and Section IV-B2, respectively.

1) Intra Prediction System Design: The H.264 intra prediction system, as a practical example, creates a prediction for an MB using only previously encoded, neighboring MBs immediately above, left, and diagonally above and left of the MB currently being encoded. Within these MBs available to be used for prediction, only the pixels bordering the MB being encoded are used to generate a prediction. Operations performed on these pixel values are generally either a replication or averaging operation—both can be easily modeled as linear transforms. The prediction modes examined (as defined by [25]) are horizontal projection, vertical projection, and DC (averaging). Other prediction methods are available in the H.264 standard but have not been explicitly referenced for conciseness.

a) Horizontal Prediction: The horizontal intra prediction mode utilizes the rightmost column of pixels from the reference block immediately to the left of the MB currently being encoded. These pixels are projected to fill each row of the prediction with the same value. In terms of a linear transformation, the elements of $\mathbf{T}$ can be expressed as

$$t_{k,l}[w] = \begin{cases} \delta\left[\left\lceil \frac{k}{N_B} \right\rceil N_B - l\right] & \text{if } w = -1 \\ 0 & \text{otherwise,} \end{cases} \tag{19}$$

where k and l are the row and column indexes for the matrix, and N_B is the horizontal dimension of an MB.

As a hypothetical example, consider the horizontal prediction transform for an MB of size $L = 2$ and a single color plane. The hypothetical MB to be transformed is represented by the raster ordered vector,

$$\hat{\mathbf{b}}_s[n-1] = \begin{bmatrix} 208 & 32 & 231 & 233 \end{bmatrix}^\mathsf{T}. \tag{20}$$

From (19), the prediction transformation for the 2×2 MB is

$$\mathbf{T}[-1] = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{21}$$

The horizontally predicted MB is

$$\begin{aligned} \mathbf{b}_p[n] &= \mathbf{T}[-1]\hat{\mathbf{b}}_s[n-1] \\ &= \begin{bmatrix} 32 & 32 & 233 & 233 \end{bmatrix}^\mathsf{T}. \end{aligned} \tag{22}$$

Assuming the pseudo-randomly generated binary generating matrix,

$$\mathbf{G} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \tag{23}$$

the stored signature associated with the stored MB (assuming no prior errors) is

$$\begin{aligned} \hat{\mathbf{s}}_s[n-1] &= \mathbf{G}\hat{\mathbf{b}}_s[n-1] \pmod{2^8} \\ &= \begin{bmatrix} 240 & 183 \end{bmatrix}^\mathsf{T}. \end{aligned} \tag{24}$$

From (4), the differential signature transform associated with the horizontal MB prediction transform is

$$\begin{aligned} \mathbf{D}[-1] &= \mathbf{G}(\mathbf{T}[-1] - \mathbf{I}) \\ &= \begin{bmatrix} 0 & 0 & -1 & 1 \\ -1 & 1 & -1 & 1 \end{bmatrix}. \end{aligned} \tag{25}$$

Using (25), the signature associated with the predicted MB is computed as

$$\begin{aligned} \mathbf{s}_p[n] &= \hat{\mathbf{s}}_s[n-1] + \mathbf{s}_d[n] \pmod{2^8} \\ &= \hat{\mathbf{s}}_s[n-1] + \mathbf{D}[-1]\hat{\mathbf{b}}_s[n-1] \pmod{2^8} \\ &= \begin{bmatrix} 242 & 9 \end{bmatrix}^\mathsf{T}. \end{aligned} \tag{26}$$

The signature can be verified by directly evaluating the signature of the calculated prediction and comparing it to the calculated signature:

$$\begin{aligned} \mathbf{s}'_p[n] &= \mathbf{G}\mathbf{b}_p[n] \\ &= \begin{bmatrix} 242 & 9 \end{bmatrix}^\mathsf{T} \\ &= \mathbf{s}_p[n], \end{aligned} \tag{27}$$

This confirms (with high probability) the prediction was computed without error.

b) Vertical Prediction: Vertical prediction for data processing is very similar to horizontal intra prediction, projecting the bottom row of pixel values down from the MB immediately above the current MB, and can be modeled as

$$t_{k,l}[w] = \begin{cases} \delta[l - 1 - M_B N_B + M_B \\ \quad - ((k-1) \bmod M_B)] & \text{if } w = -N_F \\ 0 & \text{otherwise,} \end{cases} \tag{28}$$

where M_B is the vertical dimension of an MB measured in pixels, and N_F is the frame width measured in MBs. The prediction transform for a hypothetical 2×2 MB with $w = -N_F$ is

$$\mathbf{T} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{29}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2527303, IEEE Transactions on Circuits and Systems for Video Technology

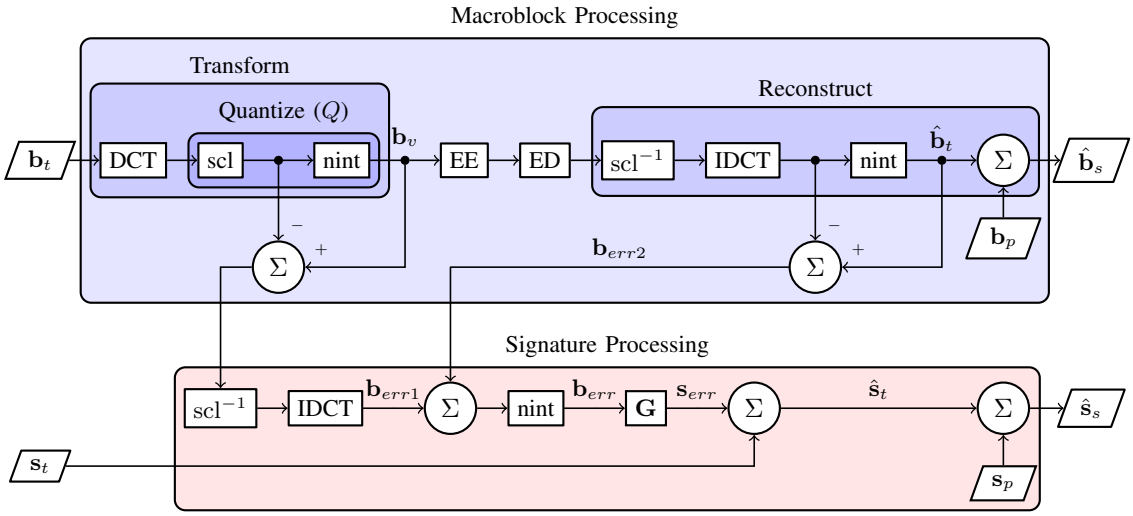

Fig. 6. Transform signature processing design. Two sources of rounding error are captured, inverse transformed, and added to the residual signature to form the reconstructed residual signature.

c) DC Prediction: The DC intra prediction mode averages the values of the immediately neighboring MBs above and to the left of the MB being encoded. The average value is used as the prediction value for all pixels in the predicted MB. The prediction transform can be expressed as

$$t_{k,l}[w] = \begin{cases} \frac{1}{cM_B}\delta\left[(l - M_B + 1) \bmod M_B\right] & \text{if } w = -1 \\ \frac{1}{cM_B}\delta\left[\left\lfloor \frac{l - M_B^2 + M_B)}{M_B} \right\rfloor\right] & \text{if } w = -N_F \\ 0 & \text{otherwise,} \end{cases} \tag{30}$$

where c is the number of MBs available for prediction (0, 1, or 2). If $w = -1$, the reference MB is immediately to the left of the current MB. If $w = -N_F$, the reference MB is immediately above the current MB. If the current MB being encoded does not have an MB to its top and/or left, c is evaluated as 1 or 0. For example, if the MB currently being encoded is the second MB in the frame, only the MB immediately to the left is available for DC prediction, yielding

$$t_{k,l}[-1] = \frac{1}{2}\delta\left[(l-1) \bmod 2\right]$$
$$\mathbf{T}[-1] = \frac{1}{2}\begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \tag{31}$$

for an MB with dimensions of 2×2. All other cases for this example would produce a zero matrix. From (31), it is shown that multiplication by the hypothetical transform will produce a vector of identical elements, equal to the mean of the second and fourth elements in the input vector. These two positions correspond with the right-most column of an MB in matrix format. Using (30), it is possible to model the DC prediction mode as a linear transform, and an associated signature transform is created by using (4).

d) Other Prediction Modes: There are a total of nine intra prediction modes defined for H.264 [25]. The other modes are similar to the three previous examples—each predicted pixel value is computed as a linear combination of stored pixel values. Therefore, for each intra prediction mode, there exists a prediction transformation matrix. This, in turn, guarantees the existence of the differential signature transformation.

2) Inter Prediction System Design: Like intra prediction, the goal of inter prediction is to produce a prediction of the current sampled MB, $\mathbf{b}_s[n]$, based on a previously encoded MB(s), $\hat{\mathbf{b}}_s[m]$. The reference MBs for inter prediction are selected from a previously encoded frame, rather than the current frame. The prediction is a group of contiguous pixels covering the same area as one MB. The group of pixels may come from up to four different MBs in a previous frame as shown in Fig. 7. The spatial relation of nine MBs are shown, with the center MB occupying the same relative frame position as the current MB being encoded. The prediction, $\mathbf{b}_p[n]$ depends on the selected frame used for prediction and the motion vector (MV), $\mathbf{v}$. In the figure, the indices for $\hat{\mathbf{b}}_s$ represent the vertical and horizontal position of the reference MB, relative to the current MB, measured in MBs.

The inter prediction computation can be interpreted as a sum of shifted reference MBs, and can, therefore, be implemented as a linear transformation. For example, in Fig. 7, to produce the upper left section of $\mathbf{b}_p[n]$, the contents of $\hat{\mathbf{b}}_s[0,0]$ should be masked to preserve only the data shaded in blue. The masked information should then be shifted up and left by $-\mathbf{v}$. Summing the masked, shifted MBs yields the prediction,

$$\mathbf{b}_p[n] = \sum_m \mathbf{T}[\mathbf{u}, \mathbf{v}]\hat{\mathbf{b}}_s[m], \tag{32}$$

where

$$u_1 = m \bmod N_F - n \bmod N_F$$
$$u_2 = \left\lfloor \frac{m \bmod M_F N_F - n \bmod M_F N_F}{N_F} \right\rfloor. \tag{33}$$

In the preceding equation, M_F is the height of each frame in the video sequence measured in MBs. The frame coordinates

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2527303, IEEE Transactions on Circuits and Systems for Video Technology

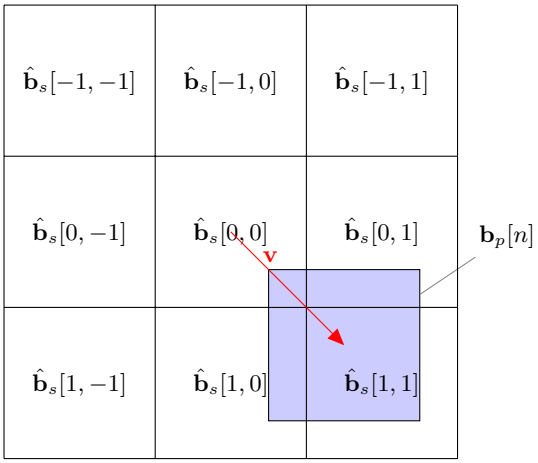

Fig. 7. Inter prediction signature processing with coordinates in terms of $\mathbf{u}$ for simplicity.

of the reference MB relative to the MB currently being processed are represented by $\mathbf{u}$. The inter prediction transform depends on the MV, $\mathbf{v}$, in addition to the relative position of each reference MB. The general linear prediction transform is defined as

$$\mathbf{T}[\mathbf{u},\mathbf{v}] = \mathbf{S}[M_B u_1 - v_1, N_B u_2 - v_2], \tag{34}$$

where $\mathbf{S}$ is the shifting and masking matrix which is defined as

$$\mathbf{S}[y,x] = \mathrm{sft}(\mathrm{repd}(\mathrm{sft}(\mathbf{I}_{M_B}, x), M_B), yM_B). \tag{35}$$

The function, $\mathrm{sft}(\mathbf{A}, b)$ shifts the rows of some matrix $\mathbf{A}$ down by b, and $\mathbf{I}_{M_B}$ is the identity matrix of size M_B. Empty rows are filled with zeros. The function, $\mathrm{repd}(\mathbf{A}, b)$ creates a block diagonal matrix of $\mathbf{A}$ repeated b times.

The function of the inter prediction transform is to shift the pixel values of each contributing reference MB into non-overlapping regions, such that the sum of all transformed reference MBs is $\mathbf{b}_p[n]$. For all reference MBs in Fig. 7 (including the ones outside the range shown) that are not partially overlapped by the prediction MB, the transform is zero and need not be computed.

As a hypothetical example, the inter prediction transform for a 2×2 MB with frame size 7×8 (in MBs) and MV, $\mathbf{v} = [\,1\ 1\,]^\mathsf{T}$ is presented. The index of the MB being encoded is $n = 65$ while the reference MB under consideration is $m = 18$. The relative location of the reference MB is directly below and right of the MB being encoded. The relative reference MB coordinates are $\mathbf{u} = [\,1\ 1\,]^\mathsf{T}$, and the transform for this case is

$$\begin{aligned}
\mathbf{T}\left[[\,1\ 1\,]^\mathsf{T}, [\,1\ 1\,]^\mathsf{T}\right] &= \mathbf{S}\left[M_B - 1, N_B - 1\right] \\
&= \mathbf{S}[1,1] \\
&= \mathrm{sft}(\mathrm{repd}(\mathrm{sft}(I_2, 1), 2), 2) \\
&= \mathrm{sft}\left(\mathrm{repd}\left(\left[\begin{smallmatrix}0&0\\1&0\end{smallmatrix}\right], 2\right), 2\right) \\
&= \mathrm{sft}\left(\left[\begin{smallmatrix}0&0&0&0\\1&0&0&0\\0&0&0&0\\0&0&1&0\end{smallmatrix}\right], 2\right) \\
&= \left[\begin{smallmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\1&0&0&0\end{smallmatrix}\right].
\end{aligned} \tag{36}$$

If the reference MB at index 18 is $\hat{\mathbf{B}}_s[18] = \left[\begin{smallmatrix}208&33\\231&233\end{smallmatrix}\right]$, the associated prediction contribution is

$$\begin{aligned}
\tilde{\mathbf{b}}_p[65] &= \mathbf{T}\left[[\,1\ 1\,]^\mathsf{T}, [\,1\ 1\,]^\mathsf{T}\right] \hat{\mathbf{b}}_s[18] \\
&= [\,0\ 0\ 0\ 208\,]^\mathsf{T} \\
\tilde{\mathbf{B}}_p[65] &= \left[\begin{smallmatrix}0&0\\0&208\end{smallmatrix}\right].
\end{aligned} \tag{37}$$

The upper left pixel in the reference MB is selected and shifted down and right. When summed with the other relevant predictions as shown in (32), the complete prediction MB is produced.

C. Transform Signature Processing

The transform system does not operate on previously encoded information. Therefore, a notation change is introduced in this section for simplicity. The index for the MBs and signatures are dropped—the index of n is implied.

Rather than attempting to perform signature processing for every subsystem within the transform and reconstruction systems, the difference between a residual MB, $\mathbf{b}_t$, and its reconstructed version, $\hat{\mathbf{b}}_t$ is observed. The transform performed by the DCT system is lossless—its inverse counterpart reconstructs the original information exactly. However, the quantization operation creates quantization error that causes the reconstructed signal to differ slightly from the original signal, and cannot be modeled as a linear transform. The proposed signature processing design for the transform system (and part of the reconstruction system) shown in Fig. 6 determines the quantization error from the MB transform and decoding operation and adds the signature associated with the error to $\mathbf{s}_t$ to produce $\hat{\mathbf{s}}_t$. The previously computed $\mathbf{s}_p$ can then be added to produce the reconstructed signature, $\hat{\mathbf{s}}_s$. Essentially, the quantization error is captured and added into the signature processing system.

There are two sources of quantization error. Both are committed to the MB data with a rounding operation shown by the nearest integer (nint) operation in Fig. 6. Without the rounding operation, there is no data loss. Therefore, a reconstructed MB can be defined in terms of quantization error by

$$\begin{aligned}
\hat{\mathbf{b}}_s &= \hat{\mathbf{b}}_t + \mathbf{b}_p \\
&= \mathbf{b}_t + \mathbf{b}_{err1} + \mathbf{b}_{err2} + \mathbf{b}_p,
\end{aligned} \tag{38}$$

where $\mathbf{b}_{err1}$ and $\mathbf{b}_{err2}$ are the fractional rounding errors added during the transform and reconstruction process shown in Fig. 6. Since adding the error to the MB signal is a linear operation, the signature generated from the quantization error can be directly added to the residual signature:

$$\begin{aligned}
\hat{\mathbf{s}}_s &= \mathbf{s}_t + \mathbf{s}_{err} + \mathbf{s}_p \\
&= \mathbf{s}_t + \mathbf{G}\left[\mathbf{b}_{err1} + \mathbf{b}_{err2}\right] + \mathbf{s}_p \pmod{2^l}.
\end{aligned} \tag{39}$$

The prediction signature, $\mathbf{s}_p$, is calculated using (14).

The same quantization error that defines the difference between $\mathbf{b}_t$ and $\hat{\mathbf{b}}_t$ also defines the difference between $\mathbf{b}_s$ and $\hat{\mathbf{b}}_s$:

$$\hat{\mathbf{b}}_s = \mathbf{b}_s + \mathbf{b}_{err} \pmod{2^l}. \tag{40}$$

Thus, it may seem practical to validate the entire encoding system by adding the quantization signature error $\mathbf{s}_{err}$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2527303, IEEE Transactions on Circuits and Systems for Video Technology

to each originally generated signature, $\mathbf{s}_s$, to validate each reconstructed MB. However, such a design leaves the MB prediction systems unprotected. Since the computed prediction is first subtracted from the sampled MB and later added during reconstruction, erroneous predictions would be masked and not detected. For this reason, the prediction system must be modeled as indicated in Fig. 5, but the transform system can be simplified as shown in Fig. 6, eliminating the need to model the nonlinear systems exactly.

D. Error Detection and Mitigation

Error detection in the proposed encoder design is performed by comparing each MB to its corresponding signature after prediction has been performed as shown in Fig. 4. The comparison point is selected to ensure protection of all systems in the video encoder, including the prediction systems. After prediction has completed for a given MB, a validation signature is created by multiplying the residual MB by the generating matrix:

$$\mathbf{s}'_t[n] = \mathbf{G}\mathbf{b}_t[n] \pmod{2^l}. \tag{41}$$

Error detection is calculated by comparing the validation signature to the residual signature using an XOR operation. The error detection indicator is defined as

$$err[n] = \mathbf{s}_t[n] \oplus \mathbf{s}'_t[n]. \tag{42}$$

An error is indicated if $err \neq 0$. Due to the cyclical nature of data in the video encoder, if an error is detected, there are multiple potential sources of the error. The error could be induced by the prediction system while computing the current MB, or the error could be a result of incorrectly processing any of the MBs used as reference for the prediction. All potential sources of the detected error must be addressed to prevent the effects of the error from propagating.

Detected errors are mitigated by preventing MBs containing errors from being used as a reference for predicting any future MBs. More conventional thinking may motivate the mitigation strategy of reprocessing the MBs possibly containing errors. However, as error rates increase, reprocessing becomes impractical as it demands many times the typical amount of processing for the same amount of information. Instead, reference MBs and the MB containing the detected error are marked as unavailable for prediction. Reference MBs are marked as unavailable because a detected error may be a result of prediction operations on the current MB or transform operations on the reference MBs. As a result, the effects of detected errors are quarantined and not allowed to propagate. Artifacts present in the decoded video sequence will be transient and will be limited to a single MB. If there are no available references from which to predict an MB, a reference-free prediction mode is selected by the encoder. In the case of H.264, this can be achieved by beginning a new slice with the MB in question and selecting the DC intra prediction mode.

Although all the MB and signature processing systems can be implemented with unreliable hardware, the XOR comparison and control system need to be implemented on reliable hardware–reliable hardware is not required for the processing of signature, in general. There are many ways of making the control system reliable, including hardening the circuit or using separate hardware. Despite the need for reliable hardware, the overwhelming majority of operations performed for video encoding can be performed on unreliable hardware. Guaranteeing the reliability of the comparator and control system is a much more manageable task compared to the entire video encoder (which is the current solution).

V. Results

A. Experimental Setup

To evaluate the effectiveness of the proposed PISP design, software based simulations were performed so that error rates could be controlled. The H.264/AVC JM Reference Software [27] was augmented to implement PISP functionality and error injection capability. Error modeling was performed by randomly flipping bits at the output of each data operation (multiply, add, etc.) with a controlled bit error rate (BER). The BER used in all experiments refers to the probability of a given bit of the output of an add operation. For all experiments, the same tests are performed on the proposed encoder design as well as the conventional design as a baseline.

Both the implemented PISP and conventional video encoders produce an H.264 protocol compliant bit-stream. The encoders can perform both intra and inter prediction, however, the intra prediction modes were limited to those listed in Section IV-B1. The inter prediction range was limited to maximum MV component magnitude of 16 with a maximum of 5 reference frames. Sub-pixel motion estimation was not supported, and the fast full method for ME was selected due to potential unexpected behavior of more complex algorithms in the presence of errors. Experiments on the conventional encoder were conducted with a variable instantaneous decoder refresh (IDR) period. Additionally, experiments for both encoder designs varied BER and QP values. The set of restricted prediction modes results in a higher than typical bit-rate, but provides an accurate baseline for comparison between conventional and PISP encoding.

For all PISP design experiments performed, the signature generating matrix, $\mathbf{G}$, was a pseudo-random binary matrix with four rows, producing a 4-dimensional vector with each element consisting of 8 bits. The probability of each matrix element being equal to 1 is 0.33, with the constraint that each column of $\mathbf{G}$ must contain at least a single 1 value. Dimensionality can be increased or decreased to modify the likelihood of error detection, however, since recovery is not a goal of the proposed design, the 4-byte signature suffices to provide adequate error detection capability for a single MB of data (384 bytes).

Six well known video test sequences were used as inputs for all experiments performed: city, container, crew, foreman, harbour, and soccer. All of the videos were limited to 300 frames in CIF, *YV12* format. All individual color samples were 8-bit integers. All experiments were performed on each of the test video sequences independently, and the statistical results of the combined experiments are presented.

The BER in all experiments was varied from 10^{-11} to 10^{-6} while processing the defined benchmark video sequences. This

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2527303, IEEE Transactions on Circuits and Systems for Video Technology

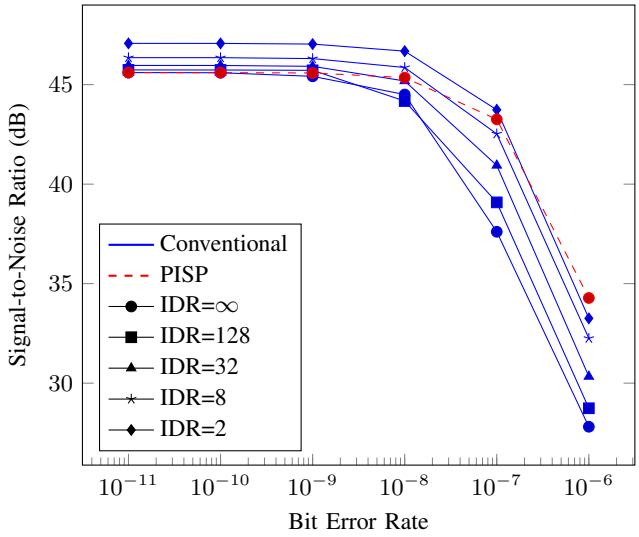

Fig. 8. Mean signal quality (relative to original image) of PISP method and conventional method for various IDR periods

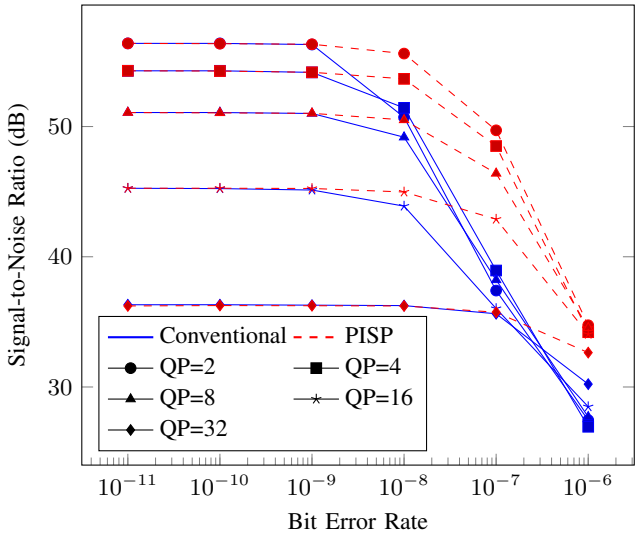

Fig. 9. Mean quality of various QP values for PISP and conventionally encoded frames with an IDR period of ∞

allows for a broad coverage of operating conditions ranging from practically no errors, to errors so frequent the hardware is practically inoperable. Error rates greater than 10^{-6} produced video output impractical for use.

B. Experimental Results

The performance of the proposed design is first evaluated against the conventional H.264 design by examining the quality of a decoded signal with a fixed QP of 16 and a varying bit error rate. The results in Fig. 8 indicate nominal performance for the conventional encoder design when BER is very low. As the BER increases, the quality of conventionally encoded sequences degrades quickly. The quality of the PISP design also degrades as the BER increases, but the rate of degradation is preferable to that of the conventional design. Reducing the IDR period of the conventional encoder produces a better quality signal, though this comes at a higher cost in terms of compression (discussed later).

To analyze the performance of the proposed design under various QP settings, the video sequences were encoded using the conventional encoder with a fixed IDR period, while varying the QP setting. The same experiments were performed on the PISP encoder, and the results are shown in Fig. 9. The quality of both conventionally and PISP encoded sequences decreases with lesser QP values or greater BER. But in each case, the quality of the PISP encoded sequences are less affected by greater BER values.

To evaluate the power performance of the proposed design, the average amount of time required to encode each individual frame was measured. The results, shown in Fig. 11, indicate the trend of more power being required for longer IDR periods, as this requires for more inter encoded frames. Each conventional encoding method tends to maintain a relatively constant workload with respect to BER. However, the PISP encoder uses less power with increasing BER. This is because there are fewer valid references from which to perform inter prediction. The PISP encoder does not spend energy performing ME for invalid MBs. Under nominal circumstances ($\mathrm{BER} \leq 10^{-11}$), the computational overhead is negligible relative to the higher IDR periods. For much higher BERs, the computation time of the PISP method drops below the fastest featured conventional method.

The relative quality improvements produced by the proposed design are offset by decreased compression performance. Fig. 12 shows the bit-rate performance (per frame) of the proposed and conventional encoders. Under near-nominal conditions, the PISP design increases the bit-rate by 2.7% compared to the conventional design, while having little impact on quality. However, with a BER of 10^{-7}, bit-rate performance remains relatively unchanged while an increase in quality of 5.7 dB is observed. Even at the very high BER of 10^{-6} with a 5.5% increase over conventional (infinite IDR period) bit-rate, the PISP design increases mean signal quality by 6.5 dB.

Mean quality measurements do not demonstrate the change of video quality over time. In Fig. 13, the quality outcome of the experiments is shown as a function of time. Without the forced memory refresh, the conventionally encoded sequences continually accumulated errors resulting in a continually decreasing quality for $\mathrm{BER} > 10^{-9}$. Although the BER determined the mean quality of the PISP encoded sequences, the quality remained more constant over time, and did not degrade as the conventionally encoded sequences did.

The effect of quality degradation over time can be observed in Fig. 10. Using the foreman sequence as a benchmark, the images in Figs. 10 (a–e) and Figs. 10 (f–j) represent regularly-spaced, decoded frames that were encoded conventionally and with PISP, respectively. The accumulation of artifacts is evident in the conventionally encoded frames, while the quality of the PISP encoded frames remains relatively consistent. The artifacts observed in the PISP encoded frames are due to errors induced while processing the currently displayed frame only. The artifacts observed in the conventionally encoded frames

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2527303, IEEE Transactions on Circuits and Systems for Video Technology

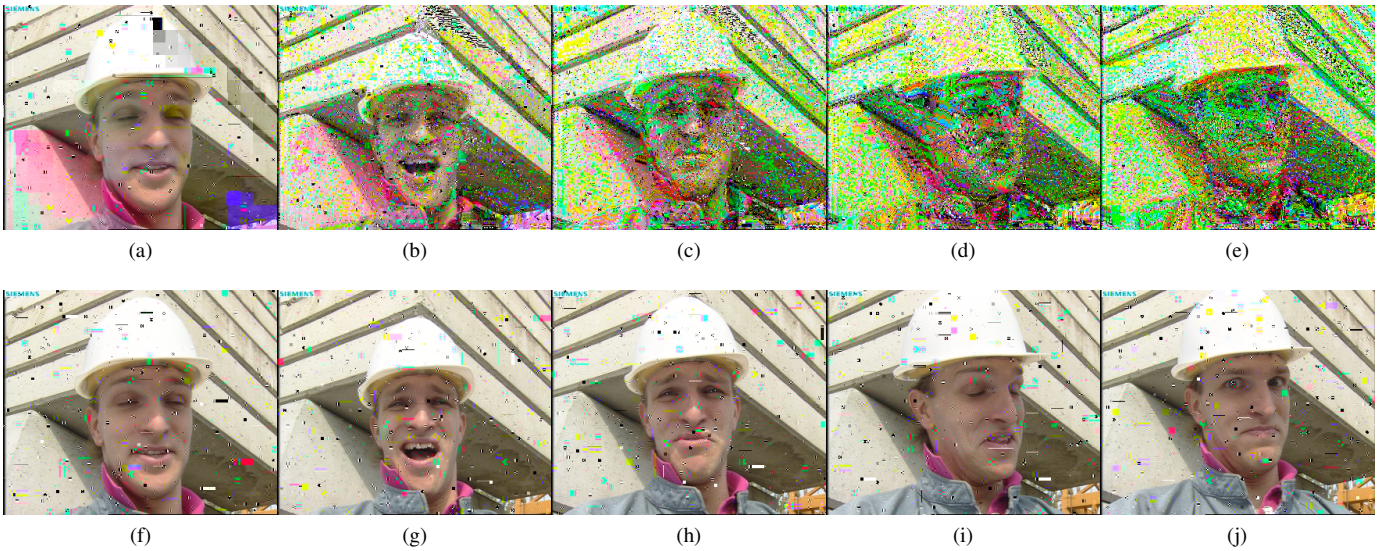

Fig. 10. Qualitative comparison of conventional inter prediction encoding (a–e) and PISP inter prediction encoding (f–j) subjected to $BER = 10^{-6}$

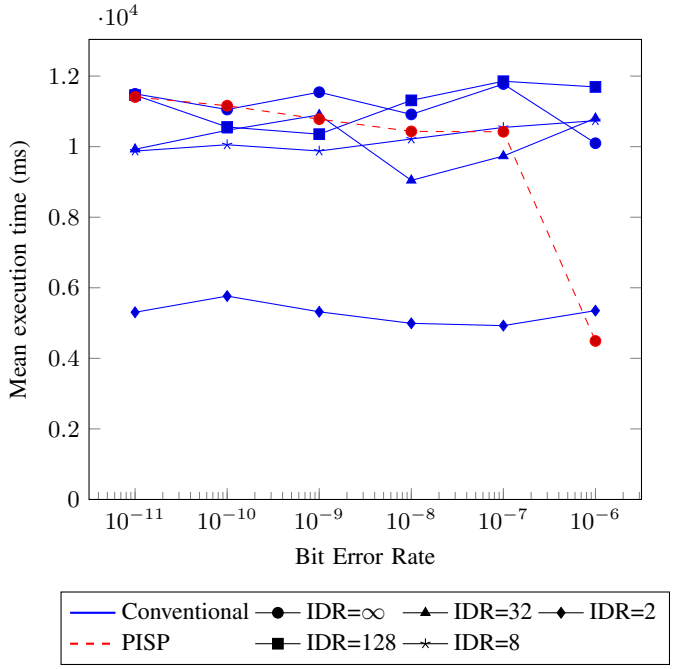

Fig. 11. Mean execution time of PISP method and conventional method for various IDR periods

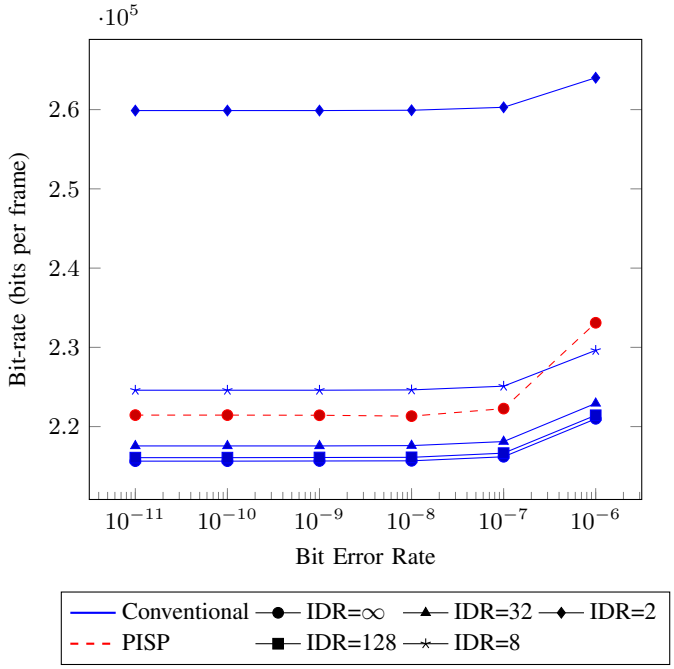

Fig. 12. Mean bit-rate of PISP method and conventional method for various IDR periods

are a result of not only errors induced while processing the displayed frame, but error induced in prior frames as well.

VI. Conclusion

Traditionally, video encoding hardware is designed for the worst case to ensure predictable operation. Parallel independent signature processing makes it possible to perform reliable video encoding on unreliable hardware. By detecting errors as they occur in the encoding hardware, the quality of the signal is allowed to gracefully degrade, rather than catastrophically failing with increasing probability of computational error. The proposed PISP design is capable of reducing the error rate in an encoded video sequence by preventing errors from propagating from one MB to another. Although some of the error mitigation methods designed in the H.264 standard are capable of increasing video quality and help to prevent decoder drift, there is an increased cost in terms of compression. The proposed method allows the encoder to adapt without making assumptions about the reliability of the hardware.

The proposed design is not without increased costs. Additional operations are required to compute the signatures used for error detection. However, the extra cost relative to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2527303, IEEE Transactions on Circuits and Systems for Video Technology

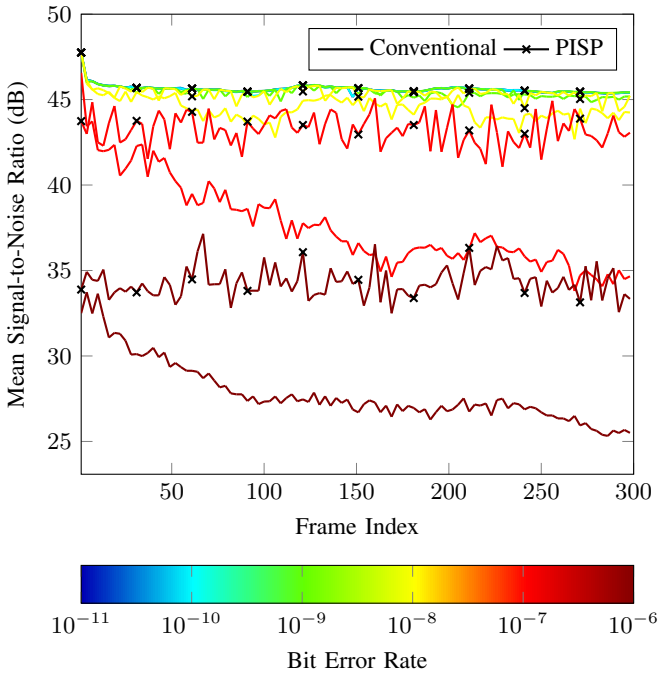

Fig. 13. Mean quality of aggregated video sequences with varying BER as a function of time with a fixed QP of 16 and a IDR period of ∞.

the whole is negligible. If the reliability of the hardware is in doubt, the proposed design can compensate for mistakes if transient artifacts in the decoded signal can be tolerated. It should also be noted that longer execution times may be observed in Fig. 11. This is due to the large number of random numbers that needed to be generated for error injection. This represents a constant factor across experiments.

The goal of this work was to demonstrate the advantages of a generalized design that can be applied to any MCHVE. Due to the complex nature of video encoders, only a subset of the prediction methods could be performed. Future goals include obtaining more precise results by including more prediction methods, and evaluating a more recent encoding standard, such as high efficiency video coding (HEVC). Since differential signature transforms can be precomputed and stored once a generating matrix is selected, the demand on memory—in terms of space and reliability—is an necessary area of future exploration.

In addition to allowing for the use of unreliable hardware, experimental results show that PISP can effectively allow video encoding hardware to better perform in error-rich environments. High-radiation environments that may cause unrecoverable artifacts in an encoded signal can be mitigated, allowing for a higher quality signal.

Errors encountered during processing do not have to be a product solely of the environment the hardware is asked to operate in. If the hardware is asked to run at a lower than ideal voltage, the same error scenario is possible. PISP allows operating at these levels to be feasible. Often, it may be desirable to preserve battery life at the expense of quality. Long term surveillance is an example of this.

Conventional video encoding implements error concealment by intra encoding frames at regular intervals. The shorter the interval, the fewer frames a potential error may be able to propagate through. If the source of errors is primarily in the video encoding hardware (as opposed to the channel), PISP can be implemented to increase the interval between intra encoded frames, allowing for greater compression.

The presented research was initially begun before the release of the latest standard from the ITU-T Video Coding Experts Group, the HEVC standard [28]. The general principle of the proposed design is transferable to the HEVC standard. Signature difference transforms can be computed for various prediction methods, and quantization error can be captured from the transform coding system to process signatures. However, it is not clear how best to segment the regions of each frame to be associated with individual signatures. Additional work needs to be done to determine if associating signatures with fixed regions across frames is best, or perhaps associating signatures with the variable sized coding blocks. Additionally, it needs to be determined whether or not the range in coding block size will lead to an increased overhead cost of maintaining signatures that will make the design impractical.

The proposed design has the potential to extend error correction ability through the channel to the decoder. Depending on the type of configuration, stored signatures may be packaged in the bit stream along with the MB, giving the decoder an extra tool for detecting errors and potentially requesting retransmission of information or invoking error concealment methods. However, this would likely necessitate changes in the H.264 standard. The proposed PISP design is implemented entirely on the encoder, and is therefore, compliant with the H.264 standard.

REFERENCES

[1] R. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Transactions on Device and Materials Reliability*, vol. 5, no. 3, pp. 305–316, 2005.

[2] J. Henkel, L. Bauer, N. Dutt, P. Gupta, S. Nassif, M. Shafique, M. Tahoori, and N. Wehn, "Reliable on-chip systems in the nano-era: Lessons learnt and future trends," in *2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, May 2013, pp. 1–10.

[3] J. F. Ziegler and W. A. Lanford, "Effect of Cosmic Rays on Computer Memories," *Science*, vol. 206, no. 4420, pp. 776–788, Nov. 1979.

[4] P. Shivakumar, M. Kistler, S. Keckler, D. Burger, and L. Alvisi, "Modeling the effect of technology trends on the soft error rate of combinational logic," in *International Conference on Dependable Systems and Networks, 2002. DSN 2002. Proceedings*, 2002, pp. 389–398.

[5] V. De and S. Borkar, "Technology and Design Challenges for Low Power and High Performance," in *Proceedings of the 1999 International Symposium on Low Power Electronics and Design*, ser. ISLPED '99. New York, NY, USA: ACM, 1999, pp. 163–168.

[6] S. Borkar, T. Karnik, and V. De, "Design and Reliability Challenges in Nanometer Technologies," in *Proceedings of the 41st Annual Design Automation Conference*, ser. DAC '04. New York, NY, USA: ACM, 2004, pp. 75–75.

[7] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: a low-power pipeline based on circuit-level timing speculation," in *36th Annual IEEE/ACM International Symposium on Microarchitecture, 2003. MICRO-36. Proceedings*, Dec. 2003, pp. 7–18.

[8] R. Rao, D. Blaauw, and D. Sylvester, "Soft Error Reduction in Combinational Logic Using Gate Resizing and Flipflop Selection," in *IEEE/ACM International Conference on Computer-Aided Design, 2006. ICCAD '06*, Nov. 2006, pp. 502–509.

[9] M. Shafique, B. Zatt, S. Rehman, F. Kriebel, and J. Henkel, "Power-efficient error-resiliency for H.264/AVC Context-Adaptive Variable Length Coding," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2012*, 2012, pp. 697–702.

[10] S. Rehman, M. Shafique, F. Kriebel, and J. Henkel, "Revc: Computationally Reliable Video Coding on unreliable hardware platforms: A case study on error-tolerant H.264/AVC CAVLC entropy coding," in *2011 18th IEEE International Conference on Image Processing (ICIP)*, 2011, pp. 397–400.

[11] J. Wells, J. Natarajan, and A. Chatterjee, "Error resilient video encoding using Block-Frame Checksums," in *On-Line Testing Symposium (IOLTS), 2010 IEEE 16th International*, Jul. 2010, pp. 289–294.

[12] W.-H. Chung, S. Paluri, S. Kumar, S. Nagaraj, and J. Matyjas, "Unequal Error Protection for H.264 Video Using RCPC Codes and Hierarchical QAM," in *2010 IEEE International Conference on Communications (ICC)*, May 2010, pp. 1–6.

[13] M. Kim, H. Oh, N. Dutt, A. Nicolau, and N. Venkatasubramanian, "PBPAIR: An Energy-efficient Error-resilient Encoding Using Probability Based Power Aware Intra Refresh," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 10, no. 3, pp. 58–69, Jul. 2006.

[14] G. Ding, H. Ghafoor, and B. Bhargava, "Error Resilient Video Transmission over Wireless Networks," in *Proceedings of the IEEE Workshop on Software Technologies for Future Embedded Systems*, ser. WSTFES '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 31–. [Online]. Available: http://dl.acm.org/citation.cfm?id=832319.838047

[15] B. Girod and N. Farber, "Feedback-based error control for mobile video transmission," *Proceedings of the IEEE*, vol. 87, no. 10, pp. 1707–1723, Oct. 1999.

[16] E. Masala, M. Bottero, and J. De Martin, "MAC-level partial checksum for H.264 video transmission over 802.11 ad hoc wireless networks," in *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, vol. 5, May 2005, pp. 2864–2868 Vol. 5.

[17] K. Lee, N. Dutt, and N. Venkatasubramanian, "EAVE: Error-Aware Video Encoding Supporting Extended Energy/QoS Trade-offs for Mobile Embedded Systems," *ACM Trans. Embed. Comput. Syst.*, vol. 11, no. 2, pp. 37:1–37:28, Jul. 2012.

[18] R. Talluri, "Error-resilient video coding in the ISO MPEG-4 standard," *IEEE Communications Magazine*, vol. 36, no. 6, pp. 112–119, Jun. 1998.

[19] Y. Wang, S. Wenger, J. Wen, and A. Katsaggelos, "Error resilient video coding techniques," *IEEE Signal Processing Magazine*, vol. 17, no. 4, pp. 61–82, Jul. 2000.

[20] H. Chung and A. Ortega, "Analysis and testing for error tolerant motion estimation," in *20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2005. DFT 2005*, 2005, pp. 514–522.

[21] H.-Y. Cheong, I. Chong, and A. Ortega, "Computation Error Tolerance in Motion Estimation Algorithms," in *2006 IEEE International Conference on Image Processing*, 2006, pp. 3289–3292.

[22] I. Polian, B. Becker, M. Nakasato, S. Ohtake, and H. Fujiwara, "Low-Cost Hardening of Image Processing Applications Against Soft Errors," in *21st IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2006. DFT '06*, 2006, pp. 274–279.

[23] G. Varatkar and N. Shanbhag, "Error-Resilient Motion Estimation Architecture," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 10, pp. 1399–1412, Oct. 2008.

[24] D. Mohapatra, G. Karakonstantis, and K. Roy, "Significance Driven Computation: A Voltage-scalable, Variation-aware, Quality-tuning Motion Estimator," in *Proceedings of the 14th ACM/IEEE International Symposium on Low Power Electronics and Design*, ser. ISLPED '09. New York, NY, USA: ACM, 2009, pp. 195–200.

[25] *Advanced video coding for generic audiovisual services*, International Telecommunication Union Std. ITU-T H.264, Feb. 2014.

[26] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[27] A. Tourapis, "H.264/AVC Reference Software - JM 18.6," 2009. [Online]. Available: http://iphome.hhi.de/suehring/tml/

[28] G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

Joshua W. Wells is a Ph.D. candidate in the School of Electrical and Computer Engineering at The Georgia Institute of Technology. He received his B.S. in Computer Engineering from The University of North Carolina at Charlotte in 2008, and his M.S. in Electrical and Computer Engineering degree from The Georgia Institute of Technology in 2011. His research interests include cross-layer, adaptive design for digital signal processing systems. He is currently researching power-efficient and error-resilient methods for video processing.

Abhijit Chatterjee is a professor in the School of Electrical and Computer Engineering at Georgia Tech and a Fellow of the IEEE. He received his Ph.D. in electrical and computer engineering from the University of Illinois at Urbana-Champaign in 1990. Dr. Chatterjee received the NSF Research Initiation Award in 1993 and the NSF CAREER Award in 1995. He has received six Best Paper Awards and three Best Paper Award nominations. His work on self-healing chips was featured as one of General Electrics key technical achievements in 1992 and was cited by the Wall Street Journal. In 1995, he was named a Collaborating Partner in NASAs New Millennium project. In 1996, he received the Outstanding Faculty for Research Award from the Georgia Tech Packaging Research Center, and in 2000, he received the Outstanding Faculty for Technology Transfer Award, also given by the Packaging Research Center. In 2007, his group received the Margarida Jacome Award for work on VIZOR: Virtually Zero Margin Adaptive RF from the Berkeley Gigascale Research Center (GSRC).

Dr. Chatterjee has authored over 425 papers in refereed journals and meetings and has 21 patents. He served as the chair of the VLSI Technical Interest Group at Georgia Tech from 2010-2012. He is a co-founder of Ardext Technologies Inc., a mixed-signal test solutions company and served as chairman and chief scientist from 2000-2002. His research interests include error-resilient signal processing and control systems, mixed-signal/RF/multi-GHz design, and test and adaptive real-time systems.