PBPAIR: An Energy-efficient Error-resilient Encoding Using Probability Based Power Aware Intra Refresh *

Minyoung Kim *minyounk@ics.uci.edu*
Hyunok Oh *hoh@ics.uci.edu*
Nikil Dutt *dutt@ics.uci.edu*
Alex Nicolau *nicolau@ics.uci.edu*
Nalini Venkatasubramanian *nalini@ics.uci.edu*
School of Information and Computer Sciences, University of California, Irvine, CA 92697-3425, USA

Error resilient encoding in video communication is becoming increasingly important due to data transmission over unreliable channels. In this paper, we propose a new power-aware error resilient coding scheme based on network error probability and user expectation in video communication using mobile handheld devices. By considering both image content and network conditions, we can achieve a fast recoverable and energy-efficient error resilient coding scheme. More importantly, our approach allows system designers to evaluate various operating points in terms of error resilient level and energy consumption over a wide range of system operating conditions. We have implemented our scheme on an H.263 video codec algorithm, compared it with the previous AIR, GOP and PGOP coding schemes, and measured energy consumption and video quality on the IPAQ and Zaurus PDAs. Our experimental results show that our approach reduces energy consumption by 34%, 24% and 17% compared with AIR, GOP and PGOP schemes respectively, while incurring only a small fluctuation in the compressed frame size. In addition, our experimental results prove that our approach allows faster error recovery than the previous AIR, GOP and PGOP approaches. We believe our error resilient coding scheme is therefore eminently applicable for video communication on energy-constrained wireless mobile handheld devices.

I. Introduction

Recent advances in technology enable mobile handheld devices to be equipped with wireless interfaces and there will be growing demand for high quality mobile multimedia communications (e.g., video telephony). However, wireless multimedia communications in the mobile handheld environment face several challenges, including high error rate, bandwidth variations, and limitations of the mobile devices such as battery lifetime constraints and the low CPU computation capability. To overcome the bandwidth limitation, there are several existing video coding techniques developed, for example, H.263 and MPEG, to compress raw video sequences to encoded bitstreams. These video encoding techniques exploit spatial and temporal correlation to achieve a high compression ratio, but they are usually unaware about the device status and the network conditions during the coding process. Therefore, multimedia data encoding requires a large amount of information, leading to high computation and communication energy consumption, and transmitting multimedia data over wireless networks can be very unreliable due to packet loss. This problem should be solved with the reasonable compression efficiency with high error resiliency considering resource constraints, which is a crucial factor for the real-time multimedia communication over error prone and lossy network using mobile handheld devices.

Video communication over unreliable networking environments is challenging since data loss and corruption from several reasons such as traffic congestion and physical channel failure affect video quality severely unless a guaranteed quality of service (QoS) is available between the source and the destination. Also, the spatiotemporal prediction encoding and variable length coding (VLC) of the source coding cause error propagation. Since spatiotemporal prediction requires the previous frame to reconstruct the current frame, a single error can lead to consec-

*A preliminary version of this work has been published in the proceedings of the First International Workshop on Services and Infrastructures for the Ubiquitous and Mobile Internet (SIUMI) 2005. This extended version includes the details on the formulation of motion vector selection. It also provides complete set of experiments for trade-off analysis on error resiliency vs. energy consumption, image quality. This work was partially supported by NSF award ACI-0204028 and ONR MURI grant N00014-02-1-0715.

utive errors in the following frames. Likewise, because of VLC, a single bit error causes the decoder to lose a synchronization point that makes the following bits useless. Therefore, a variety of techniques have been proposed to enhance the resilience of the video data encoding against the packet errors [12, 13]. The most well recognized method is to insert intra-coding[1] to mitigate the effect of error propagation in a predictive video compression algorithm [4, 5, 8, 14]. However, inserting intra-coding influences compression efficiency adversely since it tends to increase total length of the encoded bitstream. From this observation, the prior studies on error resilient video encoding mainly tried to find out a solution that maximizes bitstream robustness with low bit rate. Meanwhile, as mobile devices increasingly have video communication functionality, low power encoding has become important. Several encoding schemes have been proposed to reduce energy consumption for multimedia applications [3, 6, 7, 11]. However, these studies dealt with either error resilience or low power issues independently. We believe it is critical for both issues to be addressed together, especially in the context of energy-constrained mobile devices.

In this paper, we propose a new energy-efficient, error-resilient encoding scheme. Especially, we note the dual role of intra-coding: not only does intra-coding improve error resilience, but it also contributes to reducing encoding energy consumption since it does not require motion estimation (which is the most power consuming operation in a predictive video compression algorithm). Indeed, the system designer will therefore need to evaluate the trade-off between the error resiliency level, compression efficiency, and power consumption. In this paper, we focus our attention on these tradeoffs. Specifically, we (i) propose PBPAIR (Probability Based Power Aware Intra Refresh), a new energy-efficient and error-resilient encoding scheme, based on the network condition and the image content, (ii) implement our scheme as well as other existing error resilient encoding schemes on an H.263 codec, (iii) extensively compare with other error resilient encoding schemes in the context of error resiliency vs. encoding efficiency (both bit rate and energy consumption), and (iv) evaluate the trade-offs between the error resiliency level, compression efficiency, and energy consumption on top of real implementation platform. Our performance results indicate that PBPAIR saves as much as 17% to 34% energy compared with other error resilient techniques allowing faster error recovery than the previous approaches.

This paper is organized as follows. In the next section, we briefly review previous works on error resilient coding schemes. In Section III, we state the problem we are addressing and the proposed technique will be discussed extensively. Section IV presents experimental setup and results. In Section V we draw conclusions and comment on possible extensions of this work.

II. Error Resilient Coding

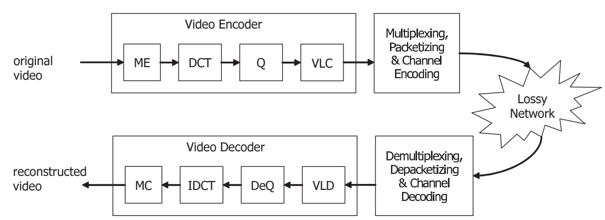

Figure 1: A typical video communication system

Figure 1 shows a typical video communication system. The original video is first compressed by a video encoder and the encoded bit stream is then multiplexed and packetized. After packetization, the bit stream usually undergoes a channel coding stage unless the network guarantees error free transmission. At the receiver end, the transmitted bit stream should be decoded to reconstruct the original video. Therefore, it is important to devise video encoding schemes that can make the compressed bit stream resilient to transmission errors since, in practice, current network environments only support lossy data transfer. Error control can be carried out at different levels from the application (codec) layer to the network transport layer. A good survey on error resilient coding techniques for real-time video communication can be found in [12, 13]. At the network transport layer, channel coding such as forward error correction (FEC) and automatic repeat request (ARQ) are applicable. However, these methods cannot guarantee complete recovery from errors and the decoder may still experience erroneous data streams. To make matters worse, these errors propagate throughout the subsequent frames since encoding is based on the difference between successive frames. To reduce these effects, the encoder should consider error-resilience

[1]For every macro block in a predictive frame (P-frame), the encoder decides whether it already knows this block from the preceding frame or whether it's completely new. In the former case, it only encodes the differences (*inter mode*). In the latter case, it encodes the whole macro block (*intra mode*). Every macro block in an intra frame (I-frame) should be encoded in *intra mode*.

and generate more robust bitstream that will not be affected by transmission losses.

The most intuitive way to produce a robust bitstream is to insert intra frames (I-frames) periodically. In this group of picture (GOP) structure, one GOP is treated as an independent decodable entity. In other words, an I-frame serves as a *refresh* which cleans up any errors that have been propagated in the video sequence. However, I-frames are usually much larger than predictively coded frames (P-frames). This leads to several transmission problems such as buffer overflow, higher delay and link congestion due to periodic peaks in the bit rate. Moreover, I-frames are much more sensitive to errors in the sense that loss of an I-frame significantly degrades the quality of the reconstructed image of the following P-frames. Techniques to overcome these problems are adaptive intra refresh (AIR) [8, 14] and progressive group of picture (PGOP) [4, 5]. AIR and PGOP scheme insert intra-coded macro blocks (MBs) to enhance the robustness of the encoded bitstream. AIR updates the specified number of MBs that have higher difference from the corresponding MBs in the previous frame while PGOP refreshes intra-coded MBs on a column-by-column basis from left to right. Both of them eliminate the need for I-frames which means the burden of refreshing is distributed throughout all the frames, thereby producing a much smoother output rate and enhanced robustness to errors.

Nevertheless, these approaches focus only on enhancement of image quality ignoring resource constraints such as power consumption. However, resource constraints need to be managed effectively while ensuring the integrity of the image quality during video communication using mobile handheld devices. Indeed, since handheld devices (e.g., PDAs and cell phones) have limited power budget of a battery that directly affects the computational resources available for the application, there is a critical need to manage video quality within this resource budget. Therefore, in this paper we propose a new error resilient encoding technique, named *PBPAIR (Probability Based Power Aware Intra Refresh)* that can run at various operating points in accordance with resource constraints.

III. PBPAIR (Probability Based Power Aware Intra Refresh)

GOP inserts an I-frame to refresh the video data while AIR and PGOP insert intra-coded MBs after the motion estimation (ME) process in Figure 1 to alleviate the effect of error propagation in a predictive video compression algorithm. AIR inserts a pre-defined number of intra-coded MBs with the highest sum of absolute differences (SAD) or mean square error (MSE) values from the ME output. Even though PGOP inserts a pre-defined number of columns of intra-coded MBs, PGOP also uses the ME output to generate stride back MBs[2] to enhance image quality. Note that AIR puts emphasis on the content awareness since it encodes the most active part of the frame while PGOP mainly pays attention to the network status since it adapts the number of columns to be encoded as intra MBs based on packet loss rate (PLR). However, probability based power aware intra refresh (PBPAIR), which we describe in the next subsections, is integrated into the ME process to determine the motion vector (content awareness); we therefore eliminate the unnecessary ME process based on PLR (network status awareness) and thereby reduce the encoding energy consumption.

III.A. The Algorithm

For quantitative analysis, we denote each macro block (MB), the probability of correctness of the corresponding MB, and the network packet loss rate as $m^k_{i,j}$, $\sigma^k_{i,j}$, and α, respectively. Consider a quarter common intermediate format (QCIF) image, a video conferencing format with each frame containing 144 lines and 176 pixels per line: this means 9×11 MBs $m^k_{i,j} (0 \leq i < 9, 0 \leq j < 11)$ with 16×16 pixels in a QCIF frame. Hence, we introduce a 9×11 matrix C_k that contains the probability of correctness $\sigma^k_{i,j}$ of the corresponding MB $m^k_{i,j}$ in the k-th QCIF video frame as follows.

$$C_k = \begin{bmatrix} \sigma^k_{0,0} & \sigma^k_{0,1} & \cdots & \sigma^k_{0,10} \\ \sigma^k_{1,0} & \sigma^k_{1,1} & \cdots & \sigma^k_{1,10} \\ \vdots & \vdots & & \vdots \\ \sigma^k_{8,0} & \sigma^k_{8,1} & \cdots & \sigma^k_{8,10} \end{bmatrix}$$

We also introduce a user-defined parameter *Intra_Th* that captures user expectation about the error resiliency level. A higher *Intra_Th* value indicates a higher user expectation about bitstream robustness.

Figure 2 illustrates the flow of our algorithm. At the beginning, we start with an error free image frame. As time goes by, PBPAIR re-evaluates the probability of correctness of each macro block to decide encoding mode and to find best matching macro block from the

[2] In order to prevent errors that may propagate across the column being refreshed, PGOP proposes to augment the refresh process to trap these propagations by refreshing the affected MBs. They call this as stride back.

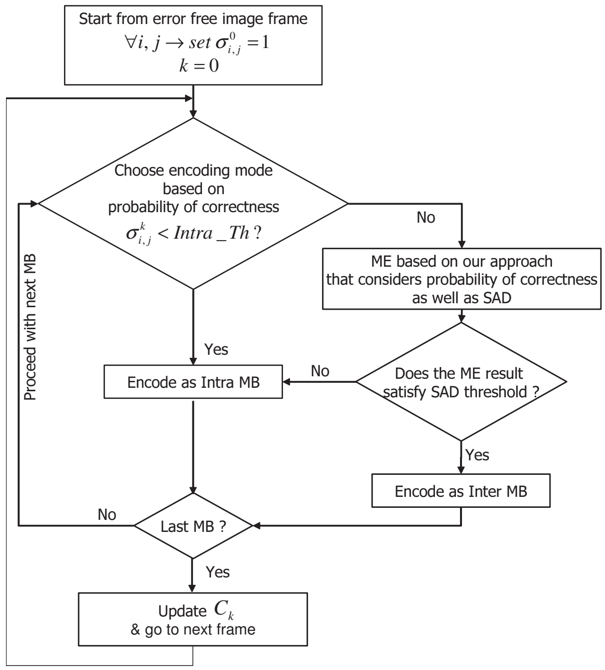

Figure 2: PBPAIR error resilient coding algorithm

previous frame. The encoding mode selection is done by comparison between probability of correctness of a MB and a given threshold value *Intra_Th*. A MB with lower probability of correctness than *Intra_Th* should be encoded as intra MB since the *Intra_Th* values can be considered as requested error resiliency level. In other words, we can skip motion estimation in this case. For a MB that is determined to be encoded as a inter-coded macro block, motion estimation based on our heuristic that considers both network condition and image content is required. Thus, our approach is integrated into the encoding process in two ways: (i) encoding mode selection and (ii) motion estimation.

Now we consider the status of network that can be expressed as packet loss rate (PLR) and user expectation (*Intra_Th*) in encoding mode selection before motion estimation (ME). The first observation here is that the image quality is guaranteed while satisfying a given constraint. However, more important contribution of this work is that PBPAIR provides various operating points in terms of image quality and resource constraints. Note that PBPAIR can operate in various manners according to a given set of constraints. If a user defines *Intra_Th* value approaches zero (meaning a user puts emphasis on compression efficiency), PBPAIR operates as if there is no error resilience feature at all. On the other hand, if user defined *Intra_Th* value equals to one, PBPAIR generates all macro blocks as intra macro block. The higher *Intra_Th* (by which a user expects higher image quality), the more intra macro blocks will be generated. Similarly, as packet loss ratio (PLR) grows, more intra macro blocks should be generated to guarantee performance requirement specified by *Intra_Th*.

We illustrate our contributions in more detail through our experimental results in Section IV. In the following subsections, we discuss our heuristic extensively. Firstly, we address our heuristic for encoding mode selection and motion estimation considering error probability of each macro-block as well as SAD based on probabilistic model. Then, we will explain how to update probability of correctness of the current frame based on that of a previous frame to re-evaluate robustness of the current encoded bitstream.

III.A.1. Encoding Mode Selection

Let us start with the first issue: How to use probability of correctness in encoding mode selection. As described in Figure 2, we can simply encode a MB as an intra-coded MB if it has lower probability of correctness than a given threshold value *Intra_Th*. The MB with low probability of correctness indicates that it is vulnerable to error propagation since that particular MB has already experienced a sufficient amount of inter-coding up to that point. We do not even have to go through motion estimation in this case. Note that this early decision improves total performance in terms of encoding time and energy, since motion estimation is very computation intensive in video compression.

III.A.2. Motion Estimation Based on Probability of Correctness

As we mentioned in Figure 2, PBPAIR not only eliminates unnecessary computation required by motion estimation (ME) with early decision based on a probabilistic model, but also considers the probability of correctness in motion vector selection. Once a MB is determined to be encoded as inter MB, then motion estimation is required. In general, the motion estimation that generates the motion vectors to determine best matching block between the previous and current frame is solely based on the sum of absolute differences (SAD). As a result, a macro block with the smallest SAD value is chosen as a reference macro block regardless of error probability caused by packet loss during transmission.

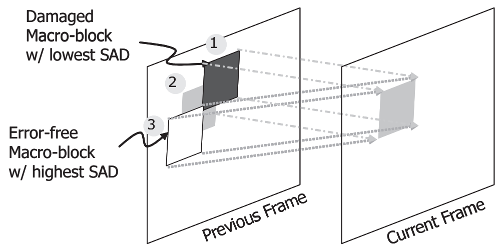

Figure 3: Motivational example for error resilient ME

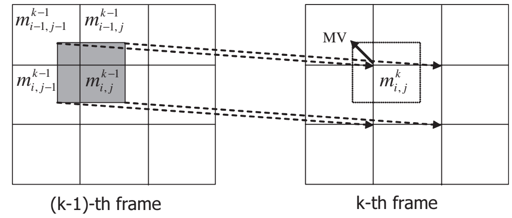

Figure 4: Motion vector (MV) and related MBs

Figure 3 illustrates the basic idea of our motion vector selection as a motivational example. Assume that there are three candidates for a reference macro block as shown in Figure 3. MB (1) has the lowest SAD value and probability of correctness among the candidates while MB (3) has the highest SAD value and probability of correctness. If we do not consider the network packet loss, we will simply choose MB (1), the candidate with the lowest SAD value. Now, assume that MB (1) is damaged during transmission. In that case, a motion vector based on only the SAD value will choose the damaged macro block as a reference block which means image quality for that macro block will be degraded. This conventional approach results in low image quality if there is an error in the previous frame. Therefore, we should consider the influence of the network packet loss in the ME process.

To take network packet loss into account, we revise the SAD formulation to propose Formula (1) which subsumes probability of correctness and image difference (SAD):

$$MV\_preference = F(C_{k-1}, SAD) = \begin{cases} weight \times normalize\{\min(\sigma^{k-1} \text{of related MBs})\} \\ +\min(\frac{SAD\_Th}{SAD}, 1) \quad \text{if } (\alpha + Intra\_Th) < 1 \\ INT\_max - SAD \quad \text{otherwise} \end{cases} \quad (1)$$

Formula (1) indicates that we choose a motion vector with higher probability of correctness and lower SAD value. This new preference value is a function of C_{k-1} and SAD between a current MB and a candidate MB where C_{k-1} indicates the matrix for the probability of correctness of the previous frame.

For example, if the motion vector equals to $(-4, -4)$, as shown in Figure 4, the related MBs of $m^k_{i,j}$ are $m^{k-1}_{i-1,j-1}$, $m^{k-1}_{i-1,j}$, $m^{k-1}_{i,j-1}$, and $m^{k-1}_{i,j}$. Then, we will use the minimum value among $\sigma^{k-1}_{i-1,j-1}, \sigma^{k-1}_{i-1,j}, \sigma^{k-1}_{i,j-1}$, and $\sigma^{k-1}_{i,j}$ since any packet loss among these MBs will degrade the image quality of $m^k_{i,j}$. Also, we need to normalize the probability of correctness since PBPAIR, as illustrated by Figure 2, simply encodes a MB with lower probability of correctness than a given *Intra_Th* as an intra coded MB. Hence, at this moment the probability of correctness $\sigma^{k-1}_{i,j}$ should be larger than *Intra_Th*. As a result, the range of $\sigma^{k-1}_{i,j}$ is now approximately $Intra\_Th < \sigma^{k-1}_{i,j} < 1 - \alpha$. A simple calculation leads to the following inequality (2).

$$0 < \frac{(\sigma^{k-1}_{i,j} - Intra\_Th)}{(1 - \alpha - Intra\_Th)} < 1 \quad (2)$$

Instead of using $\sigma^{k-1}_{i,j}$, now we use $\frac{(\sigma^{k-1}_{i,j} - Intra\_Th)}{(1-\alpha-Intra\_Th)}$ as the normalized value in Equation (1). In the case that $(\alpha + Intra\_Th) \geq 1$, the bottom formula of Equation (1) explains our heuristic that selects a lower SAD value. For all experiments, we use one as weight for correctness in Equation (1). Finding more accurate weight factor for correctness can improve our heuristic. For example, if the PLR is high, then we can emphasize the probability of correctness over the SAD by increasing weight factor.

To summarize our decision process, Algorithm 1 shows the pseudo code for PBPAIR encoding mode selection. The inequality $((SAD_{mv} - SAD\_Th) > SAD_{self})$ is used in P-frame encoding to evaluate the encoding efficiency before we actually encode the MB with generated motion vector. The term SAD_{mv} means SAD value between the current MB and the reference MB corresponding to the motion vector MV while SAD_{self} means the deviation of the pixel value in the current MB itself. If the difference between SAD_{mv} and SAD_{self} is not sufficient, then inter encoded MB probably will generate more bits than intra encoded MB. Therefore, in that case the MB should be encoded as intra MB.

III.A.3. Update Probability of Correctness

In this section, we will discuss how to generate the correctness matrix of current frame (C_k) from that of the previous frame (C_{k-1}) with a given network

Algorithm 1 Encoding Mode Selection $(C_{k-1}, \alpha, Intra\_Th)$

if $(\sigma_{i,j}^{k-1} < Intra\_Th)$ **then**
  Encoding as INTRA macro block
else
  Select motion vector based on our approach using Equation (1)
  if $((SAD_{mv} - SAD\_Th) > SAD_{self})$ **then**
    Encoding as INTRA macro block
  end if
end if

packet loss rate (α) and a motion vector. In case of *inter* macro block, the matrix for probability of correctness of k-th frame (C_k) can be calculated by Formula (3):

$$\sigma_{i,j}^{k} = (1-\alpha) \times \min(\sigma^{k-1} \text{ of related MBs}) + \alpha \times (\textit{similarity factor} \text{ between } m_{i,j}^{k-1} \text{ and } m_{i,j}^{k}) \times \sigma_{i,j}^{k-1} \quad (3)$$

The first part of Formula (3) explains the situation when the previous frame is transmitted without network packet loss in which case the probability for error-free transmission of previous frame $(1-\alpha)$ should be multiplied by the minimum probability of correctness of related macro blocks. The remaining part of Formula (3) indicates the situation when the previous frame experiences erroneous transmission such as packet lost or data corruption whose probability can be expressed by packet loss rate (PLR) α. The *similarity factor* depends on which error concealment algorithm we use at the decoder. Even when an image sample or several blocks of a sample are missing due to transmission errors, the decoder can try to estimate them based on the surrounding received samples, by making use of inherent correlation among spatially and temporally adjacent samples. Such techniques are known as error concealment techniques [13]. For instance, if we use a simple copy scheme from the corresponding MB of previous frame, we can calculate the similarity factor from SAD value between macro block $m_{i,j}^{k-1}$ and $m_{i,j}^{k}$. Note that we can easily adopt various error concealment schemes to our approach by modifying the similarity factor. For *intra* macro block, Formula (3) can be reduced to Equation (4) since this macro block will serve as a refresh.

$$\sigma_{i,j}^{k} = (1-\alpha) \times 1 + \alpha \times (\textit{similarity factor} \text{ between } m_{i,j}^{k-1} \text{ and } m_{i,j}^{k}) \times \sigma_{i,j}^{k-1} \quad (4)$$

III.B. Extension for Power Awareness

With proper interfacing mechanisms between the codec (encoder/decoder) and the network, PBPAIR can be easily modified to adjust its operations based on the network conditions and user expectation. Considering Equation (3) from Section III.A.3, the probability of correctness of the k-th frame can be approximately expected by Equation (5) if there is no similarity between the consecutive frames and every frame can be encoded as inter frame:

$$\sigma_{i,j}^{k} = (1-\alpha)^{k-1} \quad (5)$$

According to Equation (5), if the PLR (α) increases and *Intra_Th* is fixed, $\sigma_{i,j}^{k}$ decreases faster. Therefore, the PBPAIR inserts more intra macro blocks, which will result in the degradation of the encoding efficiency in terms of bit rate with less encoding energy. However, more intra macro blocks can guarantee the same level of error resiliency even though the PLR becomes higher. Furthermore, adapting (in this case, decreasing) the *Intra_Th* by the amount of the PLR increase can generate similar number of intra macro blocks. Likewise, if PLR decreases, we can increase the *Intra_Th* to encode with similar number of intra macro blocks. Note that more intra macro block represents higher error resiliency, less energy consumption, and less encoding efficiency.

This flexibility enables PBPAIR to have power awareness in the sense that it can adaptively change its operation points either to guarantee image quality within a given power constraint or to minimize power consumption with satisfying a given image quality constraint. Based on the feedback information from the network, PBPAIR can be extended to adjust *Intra_Th* parameter to maximize error resilient level within current residual energy constraint. Likewise, with a given image quality level, PBPAIR can be extended to minimize energy consumption by adapting parameters.

IV. Implementation and Evaluation

In this section, we evaluate the performance of PBPAIR through extensive experiments including power measurement on PDAs. Firstly, we compare our approach with existing error resiliency techniques: PGOP [4, 5], GOP, and AIR [8, 14]. We present two sets of experiments: (i) the effect of error resiliency with respect to energy consumption, and (ii) the variation of image quality with respect to error resiliency.

IV.A. Implementation Platform

We have implemented a PBPAIR on the H.263 encoder [9] using fixed-point arithmetic since the PDAs that we used do not have a floating point unit. We assigned 10 bits for probability of correctness (σ), inverse SAD ($\frac{SAD\_Th}{SAD}$), and *MV_preference*. We assume that a simple copy scheme is used for error concealment at the decoding side. However, as we mentioned earlier, we can easily adopt various error concealment schemes by modifying the similarity factor in Equations (3) and (4). Note that we use a uniform distribution of frame discard to generate the packet loss pattern. For simplicity, but without loss of generality, we use the frame loss rate to denote the network packet loss rate. For data transfer, we use the real time protocol (RTP) [10] and the variable-size encoded output of each frame is contained by a single packet as long as it does not exceed the maximum transfer unit (MTU).

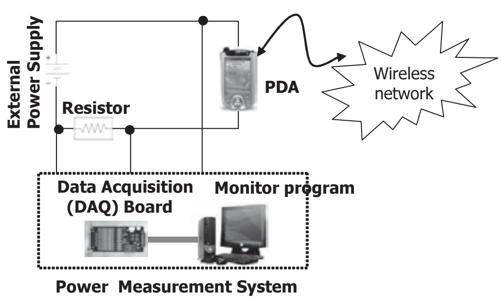

Figure 5: Experimental setup for power measurement

For power measurement, hardware platform setup in Figure 5 is used. We removed the internal battery from the PDA to measure the power consumption. All our measurements were made using a National Instruments PCI DAQ (data acquisition) board to sample the voltage drop across the resistor at 1000 samples/second. We also use two different PDAs to verify our technique: The first one is HP iPAQ H5555 with an Intel 400 MHz XScale processor with 128 MB SDRAM, 48 MB Flash ROM and integrated wireless. In the case of H5555, we installed Familiar 0.7.2 [1] with GPE environment as an operating system. The second one is Sharp Zaurus SL-5600 with an Intel 400 MHz XScale processor with 32 MB SDRAM, 64 MB Flash ROM, and external Belkin Compactflash F5D6060 wireless card. Sharp SL-5600 is powered by Linux and Java based embedded OS and the Qtopia environment for application development platform [2]. Display size for both of them is 240×320. All subsequent energy graphs depict the active energy, i.e., the total energy minus the idle energy. The results obtained allow us to derive the energy costs for encoding executions.

IV.B. Basic Comparison with Existing Error Resilient Coding Techniques

In this section, we compare existing error resilient techniques with PBPAIR to show the performance of our work. Comparison is done with GOP, AIR [8, 14], and PGOP [4, 5].

Figure 6(a) and 6(b) demonstrate the image quality on varying different parameters with several existing error resilient coding techniques, where PLR is assumed to be 10%. *NO* represents that we encode without considering any error resiliency. *PGOP-N* indicates PGOP with N-columns refresh. In other words, N-columns from left to right in a frame should be always encoded as intra MBs to enhance robustness of bitstream. On the other hand, *GOP-N* represents *I:P* ratio *I:N* where N is the number of P-frames per a single I-frame and *AIR-N* represents AIR with N intra MBs with the highest SAD values. We use the peak signal-to-noise ratio (PSNR) and number of bad pixels as image quality metric. We will discuss about image quality metric including the definitions of PSNR and bad pixels in section IV.D in more detail. We choose *Intra_Th* that gives similar compression ratio with *PGOP-3*, *GOP-3*, and *AIR-24* as shown in Figure 6(c). **Figure 6 clearly shows that PBPAIR can generate same quality of compressed image with less encoding energy consumption since our scheme skips motion estimation more frequently.** Even though PGOP also skips motion estimation for the specific MBs in the refreshing column, it still requires motion estimation for stride back MBs. This overhead will be larger with a small number of column refresh. In case of GOP, the image quality and encoding energy consumption should be similar with PGOP. In this experiment, GOP always generates a slightly smaller bitstream than other schemes because GOP generates fewer intra MBs. Hence, if we can adjust GOP to generate similar encoded file size, then the image quality and encoding energy consumption will be similar with PGOP except the overhead of stride back. Lastly, AIR consumes a similar amount of the encoding energy without any error resilient scheme since AIR decides encoding mode after motion estimation. PBPAIR reduces energy consumption due to early decision in MB mode selection and generates a robust and even bitstream against network packet loss based on the probabilistic model.

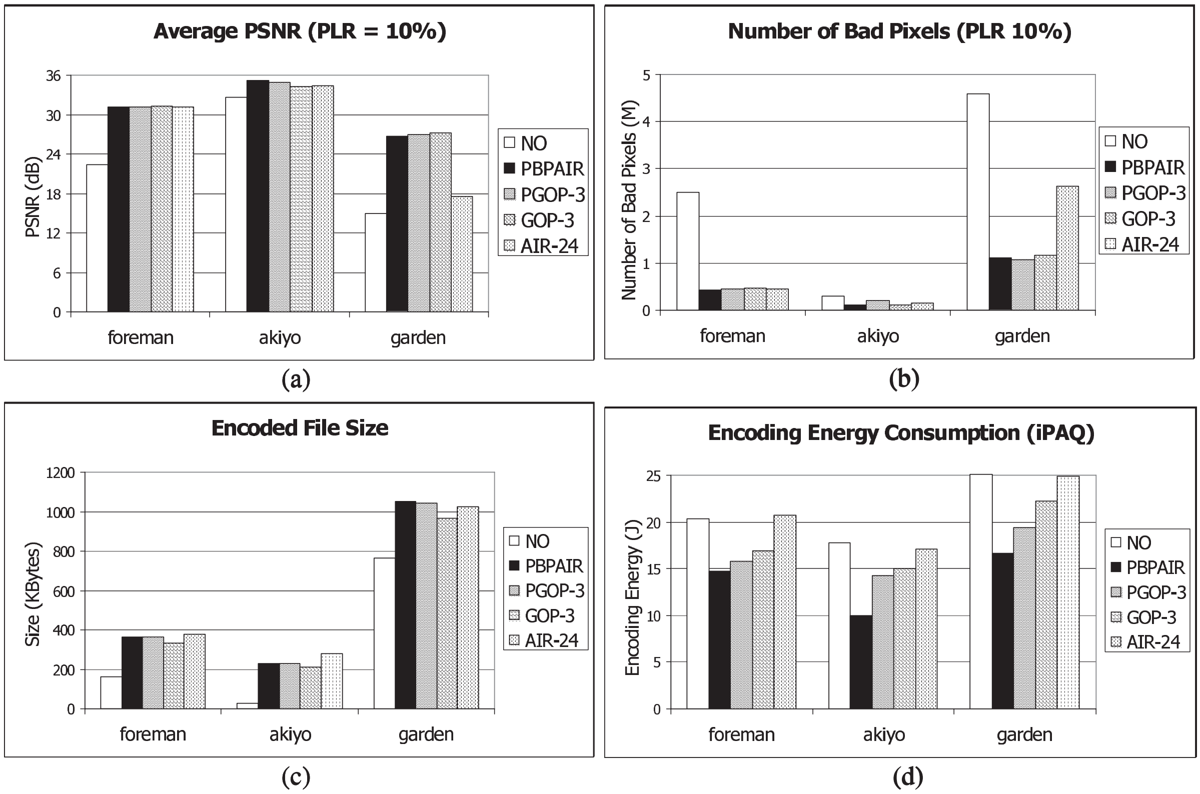

Figure 6: Comparison between PBPAIR and existing error resilient techniques such as PGOP, GOP, and AIR, where PLR is assumed to be 10%: (a) the average PSNR (b) the number of bad pixels as an image quality measure (c) the encoded file size (d) the encoding energy consumption using iPAQ as a performance measure (image source: FOREMAN.QCIF, AKIYO.QCIF, and GARDEN.QCIF, 300 frames)

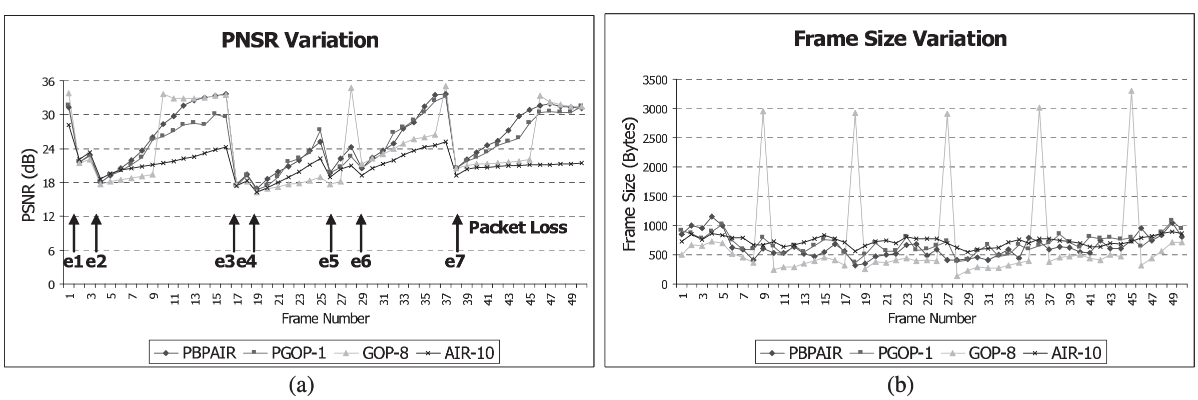

Figure 7: Comparison between PBPAIR and existing error resilient techniques such as PGOP, GOP, and AIR, where PLR is assumed to be 10%: (a) PSNR variation (b) frame size variation (image source: FOREMAN.QCIF, 300 frames)

Figure 7(a) illustrates PSNR variation according to the network packet loss represented as from $e1$ to $e7$. For comparison, we choose *PGOP-1, GOP-8*, and *AIR-10* since those schemes generate a similar size of encoded bitstream. It should be pointed out that PBPAIR recovers faster than PGOP and AIR, because our scheme not only has content awareness from the similarity factor but also has network awareness from the probabilistic model of network error. Even though GOP sometimes recovers faster than our scheme, GOP cannot guarantee rapid recovery from errors since GOP inserts an I-frame to refresh erroneous transmission. Thus, when GOP looses an I-frame due to network error, it fails to reconstruct N consecutive P-frames. The error $e7$ shows the instance of I-frame loss. Therefore, in the worst case, GOP is able to guarantee error recovery only after N frames. Moreover, Figure 7(b) shows another drawback of GOP: GOP generates an uneven bitstream that is undesirable from a communication perspective. The fact that the encoded frame size generated by GOP fluctuates severely will cause transmission problems such as buffer overflow, higher delay and link congestion due to periodic peaks in bit rate.

IV.C. Error Resilient Level vs. Energy Consumption

Figure 8(a) shows the trade-off in the number of intra macro blocks that directly affect the compressed size with a given PLR and *Intra_Th* (recall that an increased number of intra MBs results in a larger compressed bitstream as shown in Figure 8(b)). For this experiment, we encode the FOREMAN.QCIF video clip of 300 frames with a quantization coefficient of 10. The encoding results demonstrate that our probability based error resilient coding can generate an encoded bitstream with various error resiliency levels since inserting more intra MBs leads to a more robust bitstream. Also, considering *Intra_Th* is a user-defined parameter that reflects user expectation about the error resiliency level, it should be noted that our algorithm covers all possible error resiliency levels: From *Intra_Th* = 1, (which means a user wants to encode whole frames as intra MBs for maximum error resilience) to *Intra_Th* = 0 (indicating that a user wants to encode with maximum compression efficiency, without any error resilience scheme). Besides, PLR equals to zero means we can encode whole frames as P-frames. However, if the PLR approaches 1, we need more Intra MBs to guarantee required robustness. Through Figures 8(c) and 8(d), we are able to observe the trade-off between error resilient level and encoding energy consumption. We can easily expect that encoding energy consumption will be inversely proportional to the number of intra MBs, since intra coding does not require motion estimation. However, a larger number of intra blocks will result in more transmission due to the larger encoded bitstream. Considering that total energy consumption is composed of encoding energy and transmission energy, this communication overhead may affect the total energy consumption. However, the amount of possible energy reduction affected by this communication overhead is very small compared to that of the encoding tasks running on the PDA (For instance, an iPAQ consumes 0.025 $\sim$ 0.057 joules for transfer versus about 4.26 $\sim$ 23.97 joules for the encoding process.), since currently the wakeup time (from low-power sleep mode to standby mode) of the network interface with the PDAs that we used is too long to apply dynamic power management for real-time multimedia applications. It should be pointed out that, even in the situation that transmission energy is not negligible, PBPAIR still generates similar amount of bitstreams with same image quality which require similar transmission energy consumption while encoding energy in PBPAIR is the smallest as shown in Figure 6(d). Experimental results with other image samples show similar distribution except that the average number of intra MBs will be proportional to the motion intensity.

IV.D. Error Resilient Level vs. Image Quality

We now present the variation in image quality with respect to error resiliency. We use the peak signal-to-noise ratio (PSNR) as a quality metric, which is an indication of the distortion. Figure 9(a) illustrates PSNR variation with respect to different PLR and *Intra_Th*. As we explained in the previous section, a higher *Intra_Th* represents that a user requests more robust bitstreams.

We also use the number of bad pixels as a quality metric to overcome the limitation of the average PSNR since some reconstructed images with different errors have the same PNSR value. Bad pixel is defined by a pixel with

$$10 \times log_{10}\left(\frac{255 \times 255}{(PixVal_{org} - PixVal_{reconstructed})^2}\right)$$

value less than a given threshold and PSNR is defined

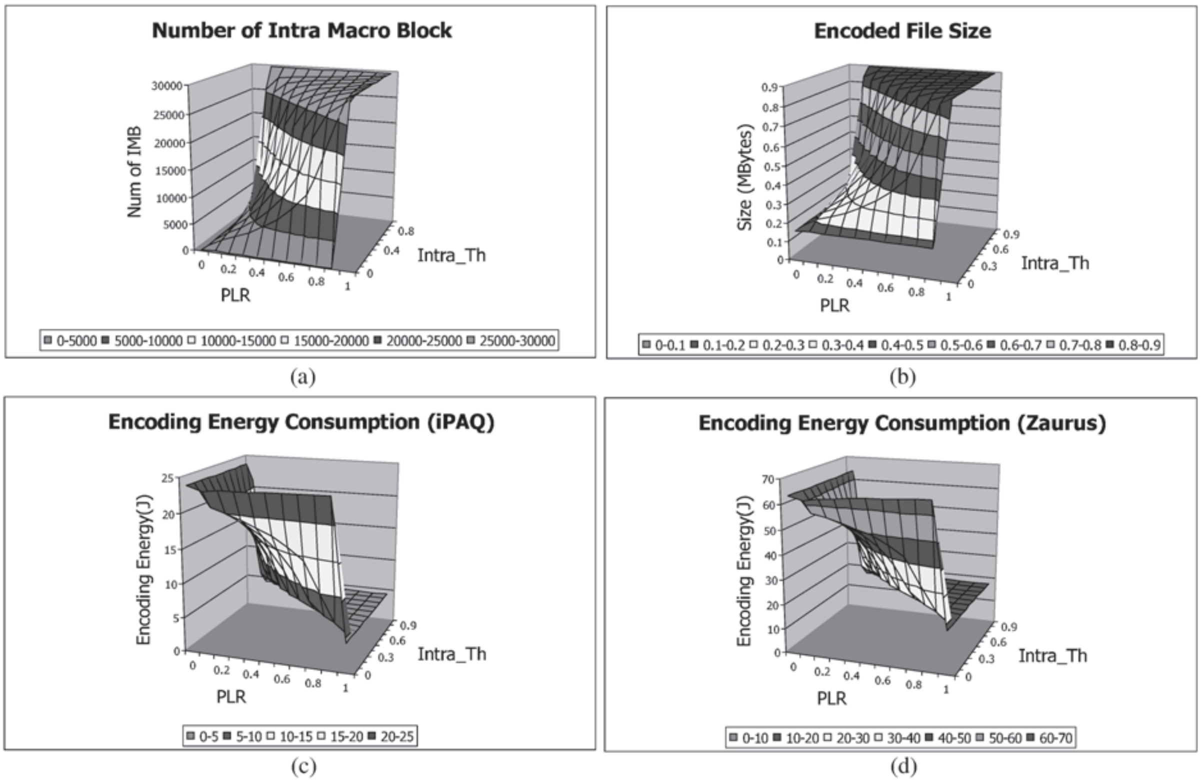

Figure 8: Trade-offs on varying PLR and *Intra_Th*: (a) the number of intra MBs (b) the encoded file size (c) the encoding energy consumption in the case of iPAQ (d) the encoding energy consumption in the case of Zaurus (image source: FOREMAN.QCIF, 300 frames)

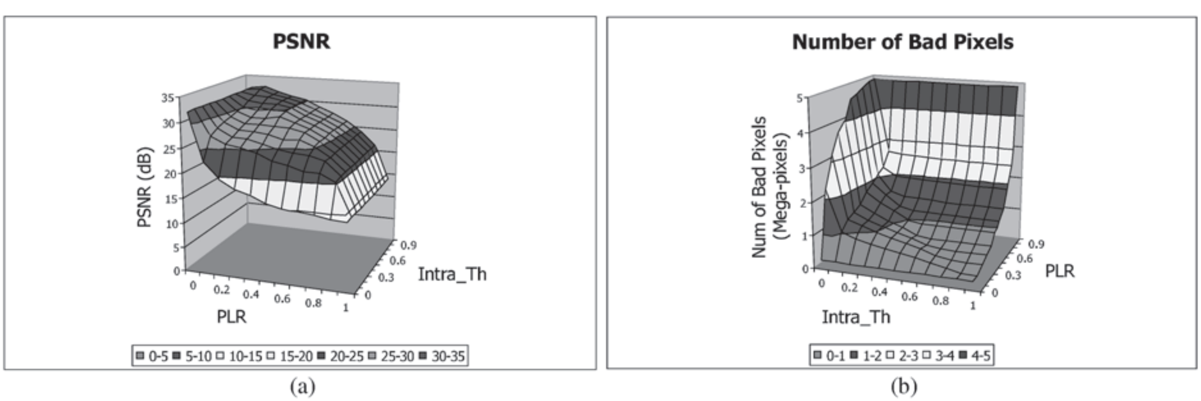

Figure 9: Error resilient level vs. image quality: (a) PSNR on varying PLR and *Intra_Th* (b) number of bad pixels on varying PLR and *Intra_Th* (image source: FOREMAN.QCIF, 300 frames)

by

$$10 \times log_{10}\left(\frac{255 \times 255}{\frac{1}{N_{pix}}\sum(PixVal_{org} - PixVal_{reconstructed})^2}\right).$$

A pixel with significant difference from the original pixel value – generated by either network error or dependency among MBs in inter frame encoding – is considered as a bad pixel. The number of bad pixels is better metric than PSNR to represent error resiliency since it counts the number of pixels which will degrade perceptive quality while PSNR depends on the reconstructed value of the bad pixels and PSNR can vary due to different encoding scheme regardless of the packet errors.

Figure 9(b) illustrates the relations among number of bad pixels (image quality), PLR (network condition), and *Intra_Th* (user expectation for robustness). Image quality varies with respect to different PLR and *Intra_Th*. As we explained in the previous section, a higher *Intra_Th* represents that a user requires more robust bitstreams. Therefore, the encoded bitstreams with higher *Intra_Th* value introduce a higher PSNR and a smaller number of bad pixels. On the other hand, if the PLR is high (meaning network is unstable), then PBPAIR can maintain the desirable image quality by advising a user to increase *Intra_Th*.

V. Conclusions and Future Work

In this paper, we proposed a new error resilient coding scheme, namely the probability based power aware intra refresh (PBPAIR), which is based on network error probability and user expectation in video communication using mobile handheld devices. By considering both image content and network condition, we can achieve fast recoverable and energy-efficient error resilient coding scheme. More importantly, we provide various operating points in terms of error resilient level and energy consumption over a wide range of system operating conditions. Our experimental results show that our approach can achieve same compression efficiency with faster recovery and reduced energy consumption by 34%, 24% and 17% compared with AIR, GOP and PGOP schemes respectively. We believe our error resilient coding scheme is therefore eminently applicable for video communication on energy-constrained wireless mobile handheld devices. Trade-offs between the power consumption and the error resilient level open a wide design space for future research subjects. Our future work will aim to design proper interfacing mechanisms between the codec and the network, so that the codec can adjust its operations based on the network conditions to maximize its resource usage. We also seek a more effective and less computationally intensive video quality measure and network packet error model for more accurate similarity factor. Cooperation with error control channel coding can be another interesting research topic since PBPAIR is independent from any other encoder/decoder side control mechanisms (e.g., rate control, channel coding, etc.). Further optimization, however, is possible if these control mechanisms are taken into consideration. Cooperation with traditional low power techniques such as dynamic voltage scaling (DVS) and dynamic frequency scaling (DFS) to explore more energy gain is also applicable as future research.

References

[1] `http://familiar.handhelds.org`.

[2] `http://www.myzaurus.com/dev`.

[3] Y. Cao and H. Yasuura. Quality-Driven Design by Bit-width Optimization for Video Applications. In *Asia and South Pacific Design Automation Conference*, 2003.

[4] L. Cheng and M. El Zarki. An Adaptive Error Resilient Video Encoder. In *SPIE Visual Communications and Image Processing (VCIP)*, July 2003.

[5] L. Cheng and M. El Zarki. Perceptual Quality Feedback Based Progressive Frame-level Refreshing For Robust Video Communication. In *IEEE Wireless Communications and Networking Conference (WCNC)*, March 2004.

[6] S. Hua, G. Qu, and S. S. Bhattacharyya. Energy Reduction Techniques for Multimedia Applications with Tolerance to Deadline Misses. In *Design Automation Conference (DAC)*, pages 131–136, 2003.

[7] C. Im and S. Ha. An Energy Optimization Technique for Latency and Quality Constrained Video Applications. In *First Workshop on Embedded Systems for Real-Time Multimedia (ESTIMedia)*, pages 18–23, 2003.

[8] ISO/IEC 14496-2. Information Technology - Generic Coding of Audio-visual Objects - Part 2: Visual, March 2000.

[9] ITU-T Recommendation H.263. Video Coding for Low Bitrate Communication, March 1996.

[10] A. Schulzrinne and S. Casner. RTP: A Transport Protocol for REAL-Time Applications. Internet draft, IETF, 1993.

[11] C. N. Tayor, S. Dey, and D. Panigrahi. Energy/Latency/Image Quality Tradeoffs in Enabling Mobile Multimedia Communication. In *Software Radio: Technologies and Services*, pages 55–66, 2001.

[12] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos. Review of Error Resilient Coding Techniques for Real-Time Video Communication. In *IEEE Signal Processing Magazine*, volume 17, pages 61–82, July 2000.

[13] Y. Wang and Q.-F. Zhu. Error Control and Concealment for Video Communication: a Review. In *Proceedings of the IEEE*, volume 86, pages 974–997, May 1998.

[14] S. Worrall, A. Sadka, P. Sweeney, and A. Kondoz. Motion Adaptive Error Resilient Encoding for MPEG-4. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1389–1392, May 2001.