Low-complexity Video Encoding for UAV Reconnaissance and Surveillance

Malavika Bhaskaranand and Jerry D. Gibson
Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA - 93106
Email: {malavika, gibson}@ece.ucsb.edu

***Abstract*—Most video compression schemes like H.264/AVC have a high-complexity encoder with a block motion estimation (ME) engine and a low-complexity decoder. However, applications such as unmanned aerial vehicle (UAV) reconnaissance and surveillance require low-complexity video encoders. Furthermore, in such applications, the majority of the motion in the video sequences is due to the movement of the UAV and the camera mounts which is known. Motivated by this, we propose and investigate a low-complexity encoder with global motion compensation and spectral entropy based bit allocation, but without block ME. The spectral entropy based bit allocation exploits latency to look ahead at data before choosing and coding the coefficients most important for retaining signal fidelity. We show that the proposed encoder achieves better quality at lower bit rates with lower quality variation than that of the H.264 encoder with ME block size restricted to 8×8 for videos typical of UAV flyovers. Compared to the H.264 encoder with 8×8 ME blocks, the proposed encoder requires fewer computations and memory accesses.**

I. INTRODUCTION

Traditional video compression schemes such as MPEG-2 and H.264/AVC have a highly complex encoder with a block motion estimation (ME) scheme and a low-complexity decoder. These compression schemes typically target applications such as entertainment video storage and playback (DVDs), video streaming, video conferencing, and entertainment broadcasting [1]. In contrast, unmanned aerial vehicles (UAVs) present video compression requirements different from the typical video compression applications. A scenario typical to UAV video compression is that the motion in the video sequences is primarily global and due to the motion of the UAV and the camera mounts. Since the motion of the UAV and the cameras are known, they can be used to efficiently estimate the global motion parameters of the video sequence. Additionally, the increasing need for more cameras in UAVs has resulted in an increase in the complexity of the processing hardware required. This problem is compounded by the "pressures on space, weight, and power (SWaP) with the desire for longer endurance units with greater functionality and lower fuel consumption" [2]. Further, lowering the cost of UAVs is also important since the failure rate is nearly 100 times that of manned vehicles [3]. Thus, it is desirable to have low-complexity video encoders in UAV payloads.

Prior research on low-complexity video encoding has mostly focused on low-complexity versions of the H.264 video compression standard and methods motivated by the theory of Wyner-Ziv distributed source coding. Low-complexity H.264 encoders [4], [5] pursue "short-cuts" to mode decisions or discard certain tool sets such as CABAC entropy coding, resulting in poorer performance. Although Wyner-Ziv video codecs [6], [7] have been shown to outperform H.264/AVC intra and sometimes even H.264/AVC 'zero-motion' coding, one of their main drawbacks is the requirement of a feedback channel and instantaneous decoding at the receiver [8]. Modifications have been proposed to avoid the feedback channel in [9], [10], but they increase the complexity of the encoder and result in a loss of quality. Recently, ISO-IEC/MPEG and ITU-T/VCEG formed the joint collaborative team on video coding (JCT-VC) with the aim to develop the next-generation video codec standard, called high efficiency video coding (HEVC). Some of the proposals to the JCT-VC have achieved visual quality similar to H.264/AVC High Profile with 20-30% bit rate reduction and lower complexity than H.264 Base Profile encoder [11]. However, none of these approaches have been tailored for UAV surveillance since they do not exploit the global motion information which is easily available.

Video compression schemes suited for UAV applications that exploit the available global motion information have also been proposed. Gong *et.al* [12] use a homography to model the global motion, merge the first intra frame and subsequent inter frame residues in a frame group into a single "big image", and code it using JPEG2000. Since JPEG2000 is not designed for encoding frame residues, there might be more efficient ways of compressing the intra frame and the residue frames. Rodriguez *et.al* [13] use the available global motion information to simplify block ME in a MPEG-4 encoder. Although their approach reduces the complexity of a standard video encoder, transmitting the global motion information instead of the motion vectors derived from it might be more efficient.

In this paper, we propose a low-complexity video encoder that uses the knowledge of the global motion in the input video sequence for global motion compensated frame prediction and spectral entropy based bit allocation for quantizer design. The spectral entropy based bit allocation scheme [14] exploits latency to look ahead at data before choosing and coding the

This research has been supported by Applied Signal Technology, Inc.

coefficients most important for retaining signal fidelity. We compare the performance of the proposed encoder to that of a H.264 encoder with ME block size restricted to 8×8 in terms of average quality at a given bit rate, quality variation across frames, and complexity. We demonstrate that the proposed encoder achieves better average quality than the H.264 encoder with 8×8 ME blocks at lower bit rates and consistently lower variation in frame quality. We also show that the proposed encoder requires fewer memory accesses and computations.

The paper is organized as follows. Section II explains the architecture of the proposed low-complexity video encoder. Section III evaluates the performance of the proposed encoder in terms of the average quality at a given bit rate, variation of video frame quality, and complexity and compares it with that of a H.264 encoder with 8×8 ME block size. Section IV summarizes the paper and discusses future improvements.

II. LOW-COMPLEXITY VIDEO ENCODER

The proposed encoder is shown in block diagram form in Fig. 1. The given video sequence of resolution $w \times h$ is split into groups of pictures (GOP) each with T frames. The first frame in each GOP is independently coded as an intra (I) frame. Then, each remaining predictive (P) frame in the GOP is predicted from the global motion compensated previous reconstructed frame. The global motion parameters are derived from the known camera motion and are not estimated from the two frames using complex algorithms. Unlike in conventional motion estimation (ME), this prediction is for the whole frame and not for individual blocks. The prediction residue is put through a spatial 2-D transform and the variances of the transform components are estimated using the coefficients. The significant coefficients that are to be coded are chosen as proposed in [15], and bits are allocated to the chosen coefficients based on the scheme developed in [14], [16]. The bits allocated are used to design scalar quantizers that operate independently on the transform components. The quantized coefficients of all the P frames in the GOP are buffered and a 1-D transform is applied along the temporal dimension. Although this transform does not decrease the entropy of the data, it improves the effectiveness of a simple entropy coder that works on individual coefficients [17].

The decoder used is "matched" to the encoder, *i.e.* it performs entropy decoding, temporal 1-D inverse transform, inverse quantization, spatial 2-D inverse transform, and global motion compensated frame reconstruction. Hence the decoder reconstructs frames that are identical to the reconstructed frames used for prediction in the encoder. Since a significant portion of the decoder is embedded in the encoder, its complexity is lower than that of the encoder. However, since there are no complexity restrictions on the decoder at the ground station, a more complex decoder could be employed [18]. With the use of a high-complexity decoder, we could allow for encoder-decoder drift thus removing the "matched" decoder embedded in the encoder and further reducing the complexity of the proposed encoder.

III. PERFORMANCE EVALUATION

For evaluating the proposed encoder, the length of each GOP is set to 8 frames. The global motion consists mostly of rotation, scaling, and translation and hence the RST global motion model is employed which is defined by

$$T_\theta(x,y) = \begin{bmatrix} \theta_3 & -\theta_4 \\ \theta_4 & \theta_3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix},$$

where (x,y) is the position of the pixel in the original frame, $T_\theta(x,y)$ the position of the pixel in the transformed frame, $[\theta_1,\theta_2]^T \in \mathbb{R}^2$ is the translation vector, and $(\theta_3,\theta_4) \in \mathbb{R}^2$ are the parameters describing the rotation and scaling. The H.264 integer transform [19] is the spatial 2-D transform applied and spectral entropy based bit allocation is used to design H.264 QMs. The I frames are coded using the H.264 intra encoding method.

The performance of the proposed encoder is compared with that of a H.264 encoder with full-pel ME restricted to 8×8 blocks around a 32×32 search range, henceforth referred to as "H.264 8x8 ME". The transform coefficients are quantized using a fixed quantization parameter (QP) and the default inter 4×4 quantization matrix (QM) as defined by the Fidelity Range Extensions (FRExt) of the H.264/AVC standard [20]. The GOP length is set to 8 and the H.264 deblocking filter is disabled.

We compare the performance of only the P frames in terms of both quality and complexity because I frames are coded identically in both encoders. The complexity of the two encoders is evaluated in terms of the size of the storage buffers required, the average number of memory accesses, and the average number of computations. The quality of the reconstructed video is evaluated using the peak signal to noise ratio (PSNR) and the Structural SIMilarity (SSIM) [21] measures. The variation of the quality across the frames is measured using the standard deviation of PSNR and SSIM. Assuming that the transform components are independent, the rate of the proposed encoder is estimated as the sum of the entropies of the individual transform components. This is the lowest rate achievable by an entropy coder that operates on the different transform components independently. We have not yet incorporated the 1-D temporal transform since it would only reduce the complexity of the entropy coder [17]. The bit rate of the "H.264 8x8 ME" encoder is computed as the sum of the rates required to code the MVs and the residue. The MV bit rate is estimated as the sum of the entropies of the MV residue components obtained after MV prediction and the residue bit rate is estimated as the sum of the entropies of the transform components after quantization.

A. Quality evaluation

In this subsection, we compare the quality achieved by the proposed encoder and the "H.264 8x8 ME" encoder in terms of the average and standard deviations of PSNR and SSIM across all P frames. The two test video sequences used are "aerial_mountain1" sequence at 368×224 resolution and

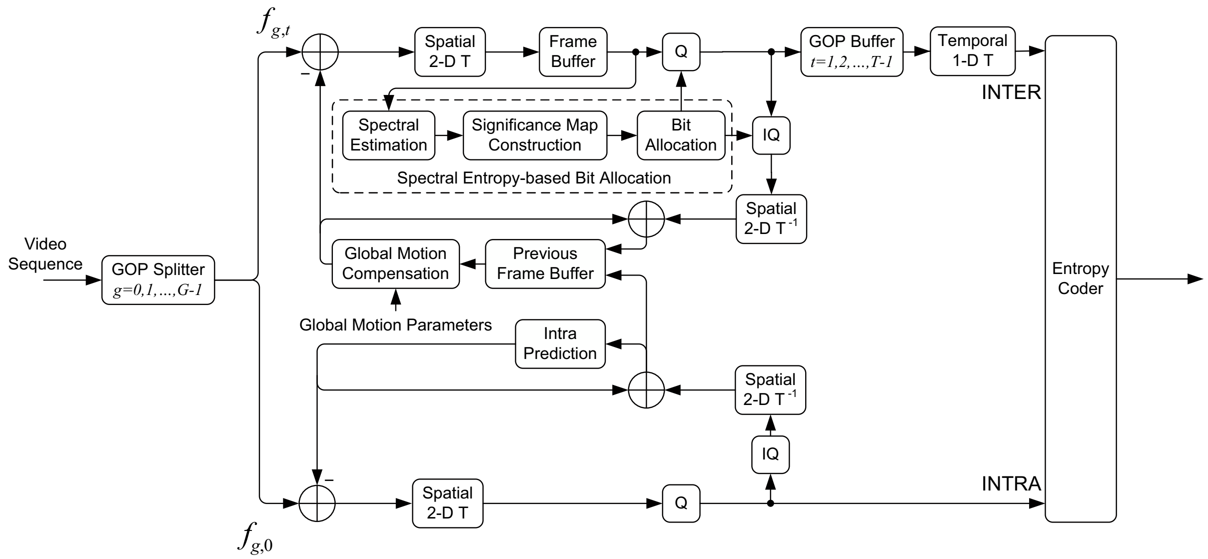
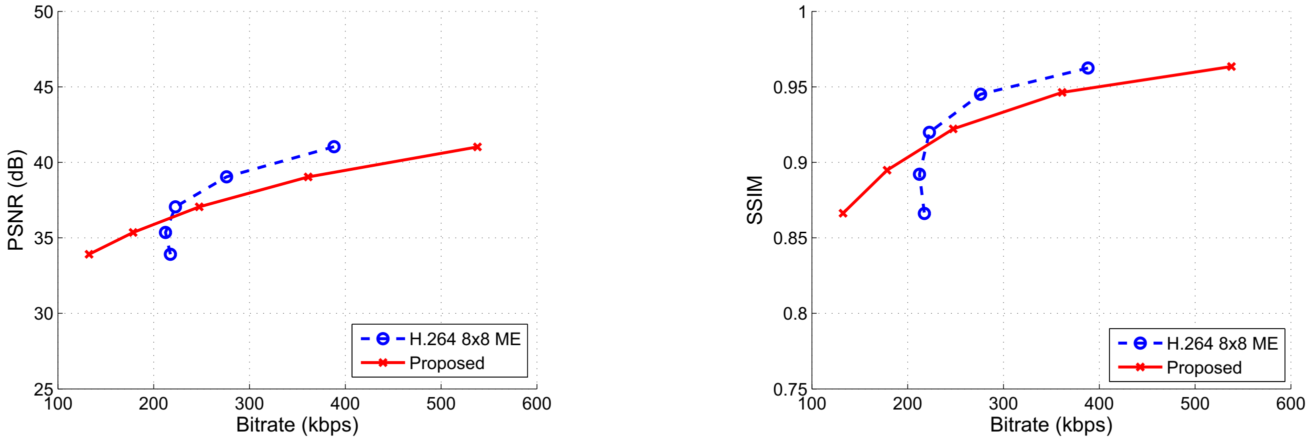
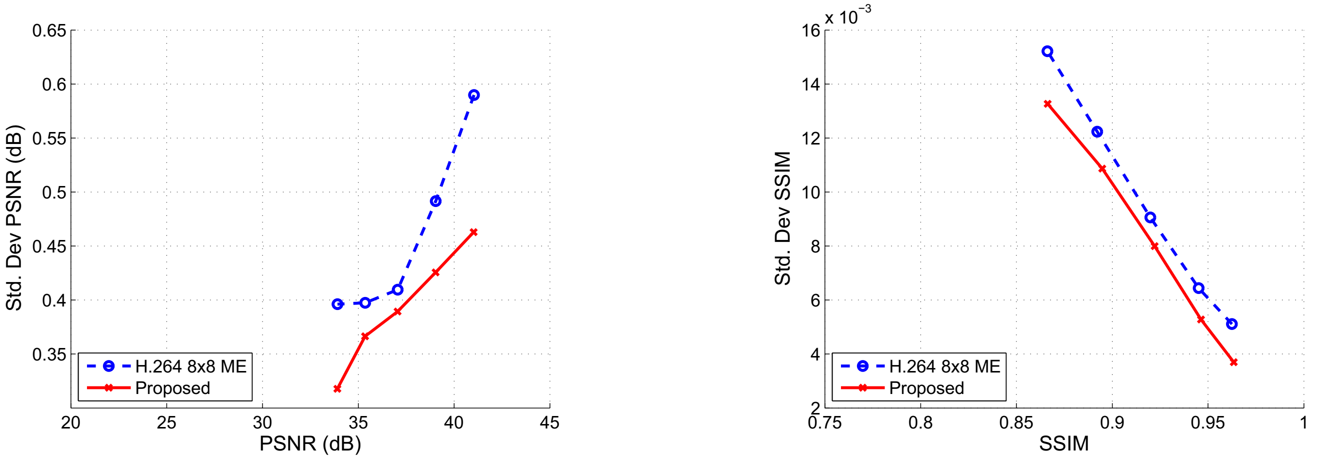


Fig. 1. Architecture for the proposed low-complexity video encoder



(a) Average PSNR vs. bit rate

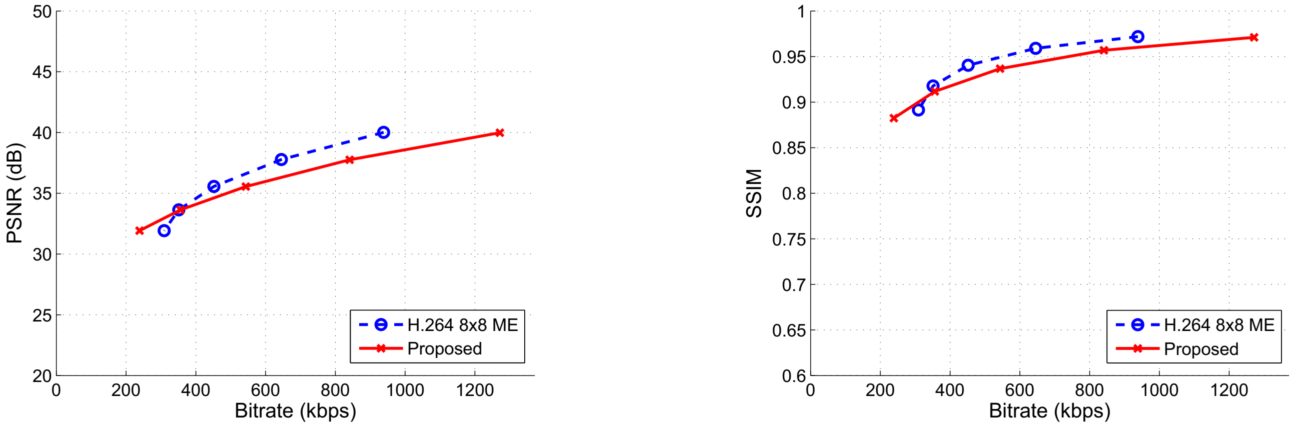
(b) Average SSIM vs. bit rate



(c) Std. deviation of PSNR vs. average PSNR

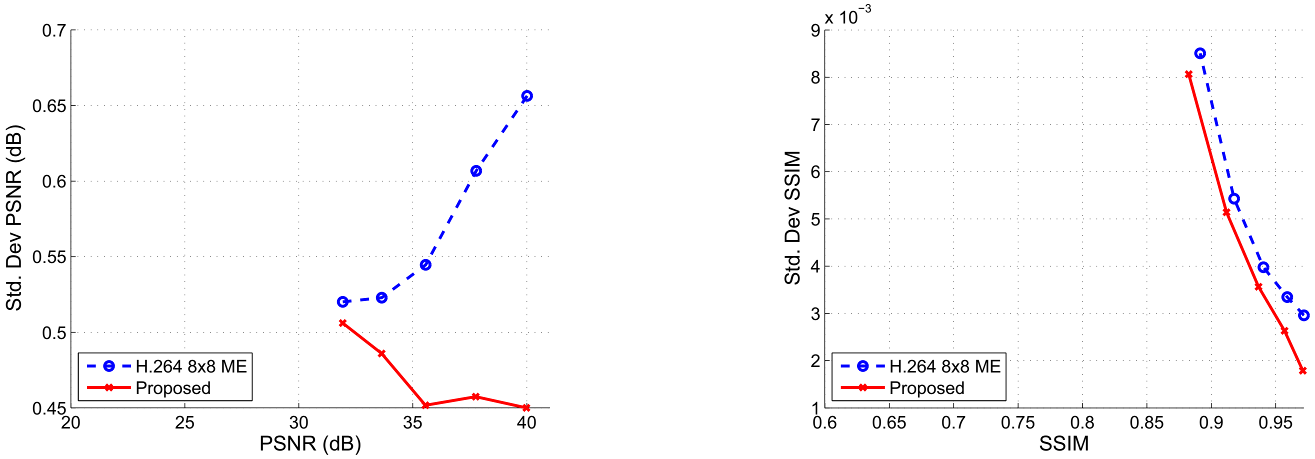
(d) Std. deviation of SSIM vs. average SSIM

Fig. 2. Comparison of performance for 368 × 224 “aerial_mountain1” sequence.



(a) Average PSNR vs. bit rate

(b) Average SSIM vs. bit rate



(c) Std. deviation of PSNR vs. average PSNR

(d) Std. deviation of SSIM vs. average SSIM

Fig. 3. Comparison of performance for 400×240 "aerial_beach1" sequence.

"aerial_beach1" sequence at 400×240 resolution. Both these sequences are typical of UAV surveillance videos and have been captured from a helicopter flying along a mountain cliff and a beach. The majority of the motion in the video sequences is due to the panning and rotation of the camera and hence is global.

Figures 2 and 3 compare the average quality and quality variation of the proposed encoder with that of the "H.264 8x8 ME" encoder for the two test videos. In terms of average quality, it can be seen that the proposed encoder performs better than the "H.264 8x8 ME" encoder at lower bit rates. This is because at lower bit rates, fewer coefficients are coded making the choice of the significant (coded) coefficients more critical and the spectral entropy based selection scheme more advantageous. Additionally, the proposed encoder consistently achieves lower quality variation across frames than the "H.264 8x8 ME" encoder as indicated by the standard deviation of PSNR and SSIM across inter frames.

B. Complexity evaluation

The complexity of the proposed and "H.264 8x8 ME" encoders are evaluated in terms of the storage required, the average number of memory accesses (reads and writes) required per macroblock processed, and the average number of computations required for processing one macroblock of data. Here the term macroblock (MB) is used to refer to an image area of 16×16 pixels. The complexity analysis is summarized in Table I with details provided in the respective subsections that follow.

TABLE I
SUMMARY OF COMPLEXITY ANALYSIS

	Proposed encoder	"H.264 8x8 ME" encoder
Memory requirements (bytes)	$12.44wh + 4w + 91$	$> 3.37wh + 4096$
Memory accesses per MB	$1536 + 512/h$	$1792 + 32768/w$
Computations per MB	2004	512∗(# candidates/block)

Table I indicates that the proposed low complexity encoder possibly requires larger memory buffers than the "H.264 8x8 ME" encoders, assuming that the H.264 based encoders require less than 4 times the memory buffers required by the respective decoders. However, in terms of the number of memory accesses, the proposed encoder does better (requires fewer) than the “H.264 8x8 ME” encoder. Also, the proposed encoder requires fewer computations than the “H.264 8x8 ME” encoder assuming ≥ 4 candidates are evaluated per block on an average for ME. It has been observed that for video sequences typical of UAV surveillance and search range of 32×32, the EPZS algorithm [22] requires more than 5 candidate searches per block on an average.

1) Memory Requirements: We compute the size of the buffers required for all the processing done prior to the entropy coding module. The memory requirements (in bytes) of the proposed encoder with RST global motion compensation are presented in Table II where $b \times b$ is the block size of the 2-D spatial transform used. Since the GOP buffer is much larger than the buffers required for intra coding, it can be reused for I frame processing. For $b = 4$, the total buffer requirement of the proposed encoder is $12.44wh + 4w + 91$ bytes.

TABLE II
MEMORY REQUIREMENTS FOR PROPOSED ENCODER

Buffer Name	Buffer Size
Previous frame buffer	$1.5 * w * h$
GOP buffer	$(T-1) * 1.5 * w * h$
Global motion compensation buffer	$4 * (w+1)$
Variance buffers for spectral entropy estimation	$b * b$
Bit allocation buffers	$2 * b * b$
Significant coefficient selection buffer	$(T-1) * (w/b) * (h/b)$
Constants	$2 * b * b + (T-1)$
Total	$T * 1.5wh + 4(w+1) + (T-1) * (wh/b^2) + 5b^2 + (T-1)$

The H.264/AVC compression standard specifies only the decoder and hence, encoder implementations greatly vary depending on their target applications. Therefore, it is difficult to compute the memory buffer requirements of a H.264 encoder. However, since decoder implementations are fairly standard, a H.264 base profile (BP) decoder typically requires $3.37wh + 4096$ bytes of storage (excluding CAVLC buffers) [23] when only 1 reference frame is used for prediction. Therefore, the “H.264 8x8 ME” encoder would require more than $3.37wh + 4096$ bytes of storage buffers. Assuming that most H.264 BP encoder implementations would require less than 4 times the storage requirements of a H.264 BP decoder, it can be concluded that the “H.264 8x8 ME” encoder requires fewer storage bytes than the proposed encoder.

2) Memory Accesses: In this subsection, we focus on the memory accesses for frame prediction and spectral entropy calculation in the proposed encoder and the block ME engine in the “H.264 8x8 ME” encoder, since these are the modules where the two encoders mainly differ. The number of memory accesses required per MB for each stage of the proposed encoder is listed in Table III where $b = 16$ is the size of the MB. In the computation of memory access during global motion compensation of the reference frame, it is assumed that every pixel in the compensated frame is bilinearly interpolated from the pixels of the input reference frame. It is reasonable to assume ≤ 2 memory accesses are required per output pixel on an average since it has been observed that a 16×16 block gets mapped to a block with ≤ 512 pixels on an average. Therefore the proposed encoder requires a total of $(6 + 2/h)b^2$ memory accesses per MB.

TABLE III
MEMORY ACCESSES PER MACROBLOCK FOR PROPOSED ENCODER

Functional stage	Memory reads	Memory writes
Input data read	$b * b$	-
Reference frame global motion compensation	$2 * b * b$	$2 * b * b/h$
2D transform + variance estimation	-	$b * b$
Quantization	$b * b$	$b * b$
Total	$4b^2$	$2b^2 + 2b^2/h$

For computing the number of memory accesses in the ME engine of the “H.264 8x8 ME” encoder, we assume n reference frames, ME blocks of size $b_{ME} \times b_{ME}$, and a search area of $S_w \times S_h$ centered about the same location as the current block. If the overlapping regions of search areas within the same reference frame are reused for adjacent blocks (level C memory reuse [24]), the number of memory accesses per MB of size $b \times b$ is given by

$$n \left[\frac{b^2}{b_{ME}^2} \right] \frac{[S_w * S_h + (w/b_{ME} - 1) * S_h * b_{ME}]}{w/b_{ME}}.$$

The assumption that the search area is centered about the same location as the current block is very restrictive and can result in degradation of coding efficiency as motion vectors (MVs) can exceed the given search range. Hence in most cases, the search area is centered about the MV predictor and a more sophisticated search area reuse algorithm such as [25] needs to be used. However, the memory reuse is much less in this case than when the search area is centered about the same location as the current block. Considering the best scenario for the H.264 encoder (search area is centered about the same location as the current block) with $n = 1$, $S_w = S_h = 32$, $b_{ME} = 8$ and $b = 16$, the total number of memory accesses required per block is $1792 + 32768/w$. Therefore, the proposed encoder requires fewer memory access per MB than the “H.264 8x8 ME” encoder.

3) Computations: Using the proposed encoder, there are significant gains in computations when compared to block ME since the ME operations have been replaced by global motion compensated frame prediction and a spectral entropy calculation. To illustrate this reduction, we express the number of computations taken by the proposed encoder for global motion compensation, differencing, and spectral entropy calculation in terms of the number of candidate blocks used during a traditional block-based motion search. We normalize this way to compare the two approaches without having to choose any particular ME algorithm, which may employ a fast search technique. The number of computations per MB for each stage in the proposed encoder is listed in Table IV.

When adding or multiplying two k-bit numbers, the complexity of addition is $O(k)$ and the complexity of multiplication is $O(k^2)$. However, it has been shown that the complexity of multiplication can be reduced to approximately $O(k^{1.5})$ [26]. Consequently, a k-bit multiplication operation can be seen as equivalent to $\sqrt{k}$ k-bit additions. Therefore,

TABLE IV
COMPUTATIONS PER MACROBLOCK FOR PROPOSED ENCODER

Functional stage	Adds	Multiplications
Reference frame global motion compensation	$3 * b * b$	$2 * b * b/h$
Frame differencing	$b * b$	-
Variance estimation	$b * b$	$b * b$
Total	$5b^2$	$b^2 + 2b^2/h$

the number of additions required per MB by the proposed encoder for global motion compensated frame prediction and spectral entropy estimation is approximately given by $(5 + \sqrt{k} + 2\sqrt{k}/h)b^2$.

For $k = 8$ and $b = 16$, frame prediction and spectral entropy computations in the proposed encoder require approximately 2004 additions per block. On the other hand, the block ME algorithm in the "H.264 8x8 ME" encoder requires $2b_{ME}^2$ additions for evaluating the sum of absolute differences (SAD) of a single candidate block. Therefore, the computational complexity of the proposed encoder is equivalent to using 4 candidates on an average per block during ME in the "H.264 8x8 ME" encoder. However, for videos typical to UAV surveillance and search range of 32×32, the EPZS algorithm [22] requires more than 5 candidate searches per block on an average. Therefore the "H.264 8x8 ME" encoder requires more computations than the proposed encoder.

IV. CONCLUSIONS

We have proposed a low-complexity encoder whose distinctive attributes are

- no block-level motion estimation,
- global motion compensated prediction with global motion parameters input from the camera mount system, and
- spectral entropy-based coefficient selection and bit allocation.

We have compared the performance of the proposed encoder with that of a H.264 encoder with 8×8 ME blocks, in terms of average R-D curves, quality variation across frames, memory storage requirements, number of memory accesses, and number of computations. We have shown that for videos typical of UAV reconnaissance, the proposed encoder achieves better R-D performance at lower bit rates and lower variation of quality across frames. We have also demonstrated that the proposed encoder requires fewer memory accesses and computations.

In this work, we have used a "matched" decoder in conjunction with the proposed encoder. The decoder complexity is therefore less than the encoder complexity. However, since in typical UAV scenarios, the decoder is housed in a ground station and no restrictions on the complexity of the decoder exist, a more complex decoder could be used [18]. The high-complexity decoder could allow encoder-decoder drift, removing the need for a "matched" decoder within the encoder and further reducing the complexity of the proposed encoder.

REFERENCES

[1] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688 – 703, jul 2003.

[2] T. Klassen, "The UAV video problem: using streaming video with unmanned aerial vehicles," *Military and Aerospace Electronics*, vol. 20, no. 7, jul 2009.

[3] J. F. Murtha, "An evidence theoretic approach to design of reliable low-cost UAVs," Master's thesis, Virginia Polytechnic Institute, 2009.

[4] Y. Tan, W. Lee, J. Tham, and S. Rahardja, "Complexity-rate-distortion optimization for real-time H.264/AVC encoding," in *Proc. Internatonal Conf. on Computer Communications and Networks*, aug 2009, pp. 1 –6.

[5] L. Su, Y. Lu, F. Wu, S. Li, and W. Gao, "Real-time video coding under power constraint based on H.264 codec," in *Proc. Visual Communications and Image Processing (VCIP)*, vol. 6508, jan 2007, pp. 1–12.

[6] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," in *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, vol. 1, nov 2002, pp. 240–244.

[7] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, "The DISCOVER codec: Architecture, techniques and evaluation," in *Picture Coding Symposium*, 2007.

[8] C. Brites, J. Ascenso, and F. Pereira, "Feedback channel in pixel domain Wyner-Ziv video coding: Myths and realities," in *Proc. of 14th European Signal Processing Conference*, sep 2006.

[9] C. Brites and F. Pereira, "Encoder rate control for transform domain Wyner-Ziv video coding," in *IEEE International Conference on Image Processing*, vol. 2, oct 2007, pp. 5–8.

[10] C. Yaacoub, J. Farah, and B. Pesquet-Popescu, "Feedback channel suppression in distributed video coding with adaptive rate allocation and quantization for multiuser applications," *EURASIP Journal on Wireless Communications and Networking*, 2008.

[11] K. Ugur, K. Andersson *et al.*, "High performance, low complexity video coding and the emerging HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1688 –1697, dec. 2010.

[12] J. Gong, C. Zheng, J. Tian, and D. Wu, "An image-sequence compressing algorithm based on homography transformation for unmanned aerial vehicle," in *International Symposium on Intelligence Information Processing and Trusted Computing (IPTC)*, oct. 2010, pp. 37 –40.

[13] A. F. Rodríguez, B. B. Ready, and C. N. Taylor, "Using telemetry data for video compression on unmanned air vehicles," AIAA Guidance, Navigation, and Control Conference and Exhibit, aug 2006.

[14] M. Bhaskaranand and J. Gibson, "Spectral entropy-based bit allocation," in *International Symposium on Information Theory and its Applications (ISITA)*, oct 2010, pp. 243 –248.

[15] W. Yang, J. D. Gibson, and T. He, "Coefficient rate and lossy source coding," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 381–386, jan 2005.

[16] M. Bhaskaranand and J. D. Gibson, "Spectral entropy-based bit allocation for given target distortion," Feb. 2011, ViVoNets Lab Internal Report.

[17] V. Goyal, "Theoretical foundations of transform coding," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 9–21, Sep 2001.

[18] J. D. Gibson, M. Bhaskaranand, Y. Liao, and S. Mangiat, "Low complexity video encoding: Trading increased delay and decoder complexity for reduced complexity encoding," Aug. 2010, ViVoNets Lab Internal Report.

[19] H. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 598 – 603, july 2003.

[20] G. J. Sullivan, P. N. Topiwala, and A. Luthra, "The H.264/AVC advanced video coding standard: overview and introduction to the fidelity range extensions," vol. 5558, no. 1. SPIE, 2004, pp. 454–474.

[21] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600 –612, april 2004.

[22] A. M. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation," in *Proceedings of Visual Communications and Image Processing (VCIP)*, Jan 2002, pp. 1069–1079.

[23] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro, "H.264/AVC baseline profile decoder complexity analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 704 – 716, july 2003.

[24] J.-C. Tuan, T.-S. Chang, and C.-W. Jen, "On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 1, pp. 61 –72, jan 2002.

[25] H. Shim and C.-M. Kyung, "Selective search area reuse algorithm for low external memory access motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 1044 –1050, july 2009.

[26] M. Fürer, "Faster integer multiplication," *SIAM Journal on Computing*, vol. 39, no. 3, pp. 979–1005, 2009.