

Resilient Video Coding using Difference Expansion and Histogram Modification

Reuben A. Farrugia

Department of Communications and Computer Engineering

University of Malta

Msida, Malta

Email: reuben.farrugia@um.edu.mt

Abstract—Recent advances in multimedia technology have paved the way to the development of several applications, including digital TV broadcasting, mobile TV, mobile gaming and telemedicine. Nonetheless, real time multimedia services still provide challenges as reliable delivery of the content cannot be guaranteed. The video compression standards incorporate error resilient mechanisms to mitigate this effect. However, these methods assume a packet-loss scenario, where corrupted slices are dropped and concealed by the decoder.

This paper presents the application of reversible watermarking techniques to facilitate the detection of corrupted macroblocks. A variable checksum is embedded within the coefficient levels and motion vectors, which is then used by the decoder to detect corrupted macroblocks which are concealed. The proposed method employs difference expansion to protect the level values while histogram modification was employed to protect the motion vectors. Unlike previous published work by the same author, this scheme does not need the transmission of side information to aid the recovery of the original level and motion vector values. Simulation results have indicated that significant gains in performance can be achieved over the H.264/AVC standard.

I. INTRODUCTION

Next Generation Networks (NGNs) will enable location independent access to multimedia services over heterogeneous devices. However, the roaming and the transmission impairments present in typical wireless networks will significantly affect the quality experienced by the end user [1], [2]. The H.264/AVC standard includes advanced error-resilient strategies to minimize the distortions in the reconstructed video [3]. Nevertheless, the standard drops all the macroblocks (MBs) contained within a corrupted slice and conceals them, even if they are not corrupted.

The Lightweight User Datagram Protocol (UDP Lite) [4] was standardized to enable partially corrupted segments to be passed to the video decoder. Nonetheless, corrupted MBs provide significant visual distortions if they are not properly detected and concealed. The authors in [5] have adopted a set of syntax and semantic rules which have only managed to detect 57% of the corrupted MBs. Error control strategies which exploit the redundancies present in compressed video was adopted in [6], [7], which managed to correct up to 70% of the corrupted MBs. The pixel-level artifact detection schemes proposed in [8], [9] manage to detect more than 95% of the corrupted MBs. However, these methods are based on heuristic thresholds and machine learning techniques, which

are hard to configure in order to ensure optimal and generalized performances.

Fragile watermarking schemes were adopted in [10], [11] to embed information that facilitates the detection of distorted MBs. However, these methods are irreversible, and reduce the quality of the transmitted video even when transmitted over an error-free channel. A reversible watermarking scheme was adopted in [12] to solve these limitations. However, this method needed the transmission of side information to be able to recover the original video quality, which increases the bandwidth required.

This paper presents the application of reversible watermarking to embed a variable checksum within each MB. The proposed method employs difference expansion and histogram modification to embed this checksum within the transformed coefficients and motion vectors respectively. Simulation results indicate that the proposed scheme achieved a significant gain in Peak Signal to Noise Ratio (PSNR) relative to the standard, especially at higher bit error rates. The proposed system provides an increase in bitrate of 6 ~ 8%, which is similar to the increase in bitrate incurred by the error resilient schemes adopted by the standard [13]. Moreover, the increased complexity can be considered as negligible. The proposed method can be easily extended to embed other information within the transmitted video.

The paper is organized as follows: Section II explains the theory behind the reversible watermarking schemes adopted in this work. The proposed system is described in more detail in Section III while the simulation results are presented in Section IV. The final comments and concluding remarks are delivered in Section V.

II. REVERSIBLE WATERMARKING

An interesting survey paper which analyzes the performance of several reversible watermarking schemes can be found in [14]. The algorithms which are most appropriate for the proposed scheme are

- Difference Expansion (DE) [15], [16]
- Reversible Contrast Mapping (RCM) [17]
- Histogram Modification (HM) [18]

The RCM method has the advantage that reversibility is ensured without the need of any side information to be transmitted to the receiver. However, the RCM transform

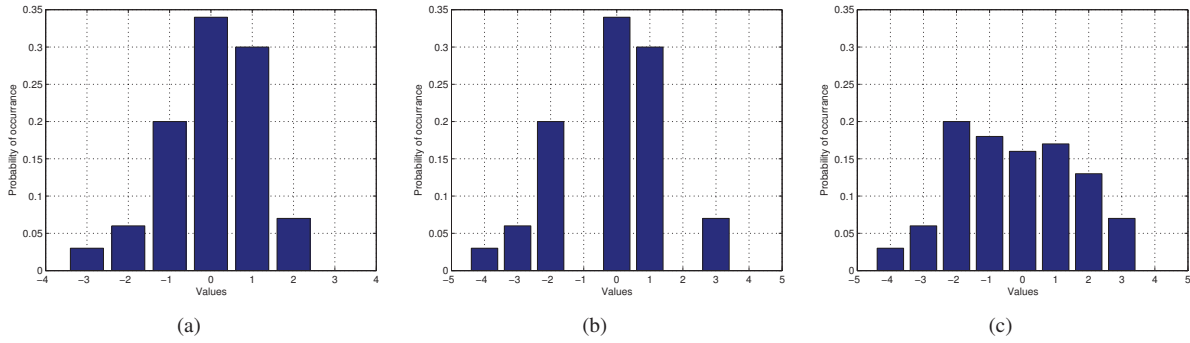


Fig. 1. Illustrative example for the Histogram Modification process (a) Original histogram (b) Histogram shifting, (c) modified Histogram

difference $\hat{x} - \hat{y} = 3(x - y)$ while the DE transform difference $\hat{x} - \hat{y} = 2(x - y)$. This analysis indicates that the DE preserves more the original statistical characteristics, and consequently preserves better the compression efficiency of the H.264/AVC standard. Another point in favor of Difference Expansion is that it provides higher embedding capacity. However, both the original Difference Expansion and Histogram Modification schemes require the transmission of side information in order to ensure reversibility.

This work presents a modified version of both Difference Expansion and Histogram Modification techniques, which do not need to transmit side information to the decoder to ensure reversibility. Therefore, given this preliminary analysis, only the modified Difference Expansion and Histogram Modification will be considered for this system. More information is provided in the following subsections and in Section III.

A. Difference Expansion

The Difference Expansion algorithm proposed by Tian [15] considers a coefficient pair $(x, y) \in \mathbb{Z}$ and computes the integer average l and difference h according to

$$l = \left\lfloor \frac{x + y}{2} \right\rfloor, h = x - y \quad (1)$$

A bit b is embedded within the difference h using the DE transformation

$$\hat{h} = 2h + b \quad (2)$$

The coefficient pairs are modified and updated using

$$\hat{x} = l + \left\lfloor \frac{\hat{h} + 1}{2} \right\rfloor, \hat{y} = l - \left\lfloor \frac{\hat{h}}{2} \right\rfloor \quad (3)$$

The decoder extracts the expanded coefficient pair (\hat{x}, \hat{y}) and computes (1) to derive the integer average l and expanded difference \hat{h} . It is important to mention that the integer average l is the same at both encoder and decoder, while the difference \hat{h} contains the embedded bit. The binary value b can be easily extracted at the receiver by taking the least significant bit value from \hat{h} , while the original difference can be recovered using

$$h = \left\lfloor \frac{\hat{h}}{2} \right\rfloor \quad (4)$$

The original coefficient values are recovered using (3), where the recovered difference h is used instead of the expanded difference \hat{h} .

Both the difference expansion proposed by Tian [15] and its extension [16] need to transmit side information in order to avoid rounding errors and ambiguities at the decoder. These ambiguities are present because both algorithms are computed in the pixel domain, where the intensity values of both coefficient pairs $(x, y) \in [0, L]$. However, these ambiguities are completely avoided if the coefficients considered for embedding $(x, y) \in \mathbb{Z}$ are not bounded to any particular range of values. Further detail will be provided in Section III.

B. Histogram Modification

The Histogram Modification algorithm adopted in this paper is similar to the one presented in [18]. The computation of Histogram Modification can be explained through an illustrative example provided in Fig. 1. Consider that the original dataset is Laplacian distributed (Fig. 1(a)) and consider L_1 and L_2 to represent the values which have the highest and second highest probability of occurrence. In this example $L_1 = 0$ and $L_2 = +1$. In order to get the space required for embedding, a coefficient x is shifted using

$$\hat{x} = \begin{cases} x - 1 & \text{if } x < \min(L_1, L_2) \\ x + 1 & \text{if } x > \max(L_1, L_2) \end{cases} \quad (5)$$

where the resulting histogram is illustrated in Fig. 1(b). It can be immediately noticed that bin values -1 and $+2$ are available for watermarking embedding. The binary bit b is embedded within the coefficients x whose values equal to L_1 or L_2 , which are modified according to

$$\hat{x} = \begin{cases} x - 1 & \text{if } x = \min(L_1, L_2) \text{ and } b = 1 \\ x + 1 & \text{if } x = \max(L_1, L_2) \text{ and } b = 1 \\ x & \text{if } b = 0 \end{cases} \quad (6)$$

The resulting modified histogram is illustrated in Fig. 1(c). The receiver receives the modified coefficients \hat{x} and extracts the embedded bit b using

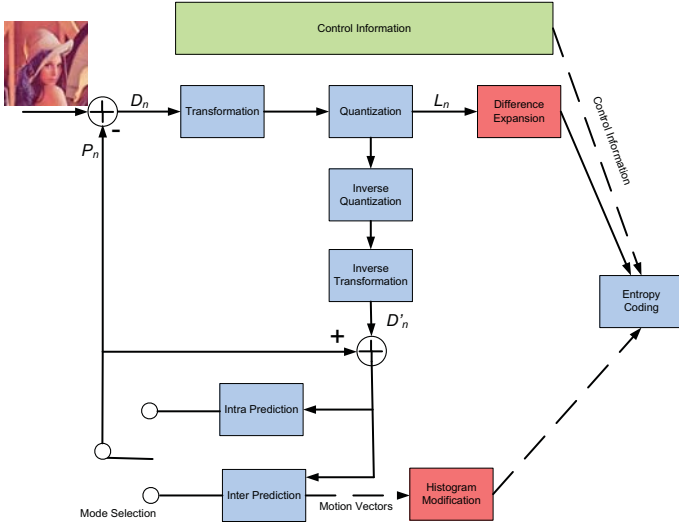


Fig. 2. Schematic diagram of the proposed system

$$b = \begin{cases} 0 & \text{if } \hat{x} \in L_1, L_2 \\ 1 & \text{if } \hat{x} \in \min(L_1, L_2) - 1, \max(L_1, L_2) + 1 \end{cases} \quad (7)$$

The original coefficient values are then recovered by

$$x = \begin{cases} \hat{x} + 1 & \text{if } \hat{x} < \min(L_1, L_2) \\ \hat{x} - 1 & \text{if } \hat{x} > \max(L_1, L_2) \\ \hat{x} & \text{if } \hat{x} \in L_1, L_2 \end{cases} \quad (8)$$

It is important to notice here that the capacity of the Histogram modification process is dependent on the probability of occurrence of values L_1 and L_2 . The capacity of the watermarking system will be compromised if for some reason, the most probable symbols cannot be employed for watermark embedding.

III. SYSTEM OVERVIEW

Fig. 2 illustrates the schematic diagram of the proposed system, which is based on H.264/AVC. The proposed scheme includes two distinct watermarking schemes prior entropy coding, which is a lossless process. The difference expansion was used to embed information within the residual information, while histogram modification was used to embed information within motion vectors. More information about these processes is provided in subsections III-A and III-B.

The proposed algorithm is used to protect both residual information and motion vectors. The number of bits that can be embedded within a macroblock is derived using

$$M = \min(M_{mv} + M_c, M_{\max}) \quad (9)$$

where M_{mv} represents the number of motion vectors that can be used for embedding (see subsection III-A), M_c represents the number of level values that can be used for embedding (see subsection III-B) and M_{\max} represents the maximum number of bits that can be embedded within a macroblock. The

proposed system embeds a variable length checksum within each macroblock using

$$K = K_\alpha \bmod M \quad (10)$$

where K_α is computed using

$$K_\alpha = \sum_{i \in N_{mv}} (mv_{x,i} + mv_{y,i}) + \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} |c_{j,k}| \quad (11)$$

where N_{mv} represent the number of motion vector pairs, $(mv_{x,i}, mv_{y,i})$ represent the i^{th} motion vector pair, N represents the macroblock width or height in pixels, and $c(j, k)$ represents the intensity level of the residual quantized coefficients at coordinates (j, k) . The first M_{mv} bits of message K are embedded within the motion vectors while the remaining bits are embedded within the level values.

The decoder receives the modified motion vectors and residual information, from which the embedded message K is extracted. The original motion vectors and coefficient level values are recovered using the methods described in Section II. The variable checksum is then computed using (10), and compared with the extracted message K . A macroblock is considered as corrupted if the checksum values do not match.

A. Reversible Watermarking within Motion Vectors

The P-macroblocks and B-macroblocks, which are used for temporal prediction, use motion vectors to aid reconstruction of the reference frame at the decoder. H.264/AVC predicts the motion vector using neighboring motion vectors, and transmits the residual motion vector mv_d , which follows a zero-mean Laplacian distribution, where most of the energy resides within the interval $[-167, +167]$. The range of the residual motion vectors is bounded based on the level and profile adopted by the application, and therefore Difference Expansion needs to transmit side information to avoid ambiguities.

Histogram modification does not suffer from ambiguous pairs and has the additional advantage that it increases the dynamic range by only ± 1 , which has a minimal effect on compression efficiency. The authors in [18] transmit side information, which contributes to an increase of 4% in bitrate to signal the L_1, L_2 and other parameters to the decoder. This work avoids the transmission of side information by exploiting the fact that the distribution of mv_d follows a zero-mean Laplacian distribution, and therefore the most probable mv_d values are 0 and ± 1 . This work considers $L_1 = 0, L_2 = +1$. The number of bits that can be embedded within a motion vector M_{mv} is equivalent to the frequency of occurrence of L_1 and L_2 .

B. Reversible Watermarking within Residual Information

The quantized residual coefficients c are used to embed the remaining $M - M_{mv}$ bits. As in the case of mv_d , the coefficients c are zero-mean Laplacian distributed, where most of the coefficient values fall within the range $[-15, +15]$. This range is not bounded and the coefficient pairs can be expanded

without causing ambiguities. The unbounded nature of the residual coefficients c eliminates the need to transmit side information to attain reversibility. Moreover, the difference h is also zero-mean Laplacian distributed, thus the distribution of the modified coefficients will be similar to the original distribution, reducing the effect on compression efficiency. These desirable properties, together with the need for high capacity suggest the employment of Difference Expansion to embed information within the residual coefficients.

The Difference Expansion method adopted in this paper exploits the fact that a macroblock is divided into sixteen 4×4 blocks of pixels, which are transformed and quantized to form c . Therefore, the 16×16 block of quantized coefficients c can be grouped into a set of sixteen transform blocks $t_{m,n}$ according to

$$t_{m,n} = \begin{pmatrix} c_{4m,4n} & c_{4m+1,4n} & c_{4m+2,4n} & c_{4m+3,4n} \\ c_{4m,4n+1} & c_{4m+1,4n+1} & c_{4m+2,4n+1} & c_{4m+3,4n+1} \\ c_{4m,4n+2} & c_{4m+1,4n+2} & c_{4m+2,4n+2} & c_{4m+3,4n+2} \\ c_{4m,4n+3} & c_{4m+1,4n+3} & c_{4m+2,4n+3} & c_{4m+3,4n+3} \end{pmatrix}$$

For simplicity, let's denote the transformed block

$$t_{m,n} = \begin{pmatrix} t_{m,n}(0,0) & t_{m,n}(1,0) & t_{m,n}(2,0) & t_{m,n}(3,0) \\ t_{m,n}(0,1) & t_{m,n}(1,1) & t_{m,n}(2,1) & t_{m,n}(3,1) \\ t_{m,n}(0,2) & t_{m,n}(1,2) & t_{m,n}(2,2) & t_{m,n}(3,2) \\ t_{m,n}(0,3) & t_{m,n}(1,3) & t_{m,n}(2,3) & t_{m,n}(3,3) \end{pmatrix}$$

where $t_{m,n}(0,0)$ represents the DC coefficient and $t_{m,n}(3,3)$ represents the highest frequency component of the transformed block. The block is zigzag scan ordered to generate an array of coefficients

$$z_{m,n} = [t_{m,n}(0,0), t_{m,n}(1,0), t_{m,n}(0,1), \dots, t_{m,n}(3,3)]$$

The remaining $M - M_{mv}$ bits of message K are embedded within the coefficient pairs $(z_{2m,n}(k), z_{2m+1,n}(k))$ for $m = 0, 1; n = 0, \dots, 3$, and $k \in 0, \dots, 15$, where k denotes the frequency band. In order to preserve compression efficiency, only coefficient pairs where at least one coefficient is nonzero is expanded using (3). The number of expandable coefficient pairs M_c is therefore easily computed by deriving the number of coefficient pairs which have at least one nonzero value. In case that $M_c > M - M_{mv}$, the $M - M_{mv}$ coefficient pairs with the smallest integer average l are used for embedding.

C. Signalling and Video Transmission

The proposed system needs that partially corrupted Network Abstraction Layer Units (NALUs) are passed to the decoder, which will then adopt the reversible watermarking schemes to detect the corrupted macroblocks that need to be concealed. The UDP protocol is normally used to transmit video information. However, the UDP protocol is sub-optimal for wireless video transmission, since it will drop all the macroblocks contained within a corrupted packet, even uncorrupted macroblocks.

TABLE I
INCLUDED SEQUENCE PARAMETER SET SYNTAX

Field	Descriptor
watermarking_enabled_flag	$u(1)$
if(watermarking_enabled_flag) {	
M_{max}	$u(4)$
L₁	$u(4)$
L₂	$u(4)$
}	

Fig. 3 illustrates the UDP Lite protocol which is used to encapsulate the video related NALUs. The UDP Lite is similar to UDP, where UDP Lite replaces the Length field of UDP with a Coverage field. The Coverage field is used to specify the number of bytes, starting from the first byte of the UDP Lite header, that are sensitive to errors. The UDP Lite header and the IP pseudo-header are always verified by the checksum. Moreover, since both the NALU header and Slice header contain sensitive information they are both protected by the partial checksum. On the other hand, the payload, which basically includes the macroblocks, is considered as insensitive information and is not protected by the partial checksum. Therefore, corrupted macroblocks are passed to the application layer and the decoder can exploit the information hidden within both motion vector and residual information to detect the corrupted macroblocks.

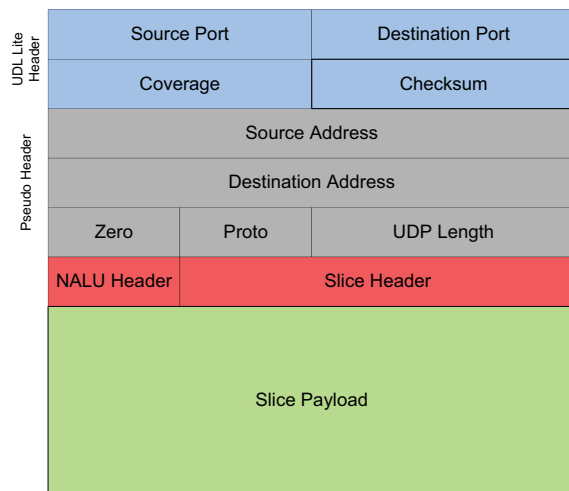


Fig. 3. UDP Lite Header and Payload

To increase the flexibility of the proposed error resilient strategy and to minimize its effect on compression efficiency, the configuration parameters needed by the decoder were transmitted using the Sequence Parameter Set (SPS). The SPS are only transmitted once with every Instantaneous Decoder Refresh (IDR) frame and are considered to be transmitted using the reliable Transmission Control Protocol (TCP). Table I illustrates the syntax included within the SPS adopted by H.264/AVC, where $u(n)$ represents unsigned fixed length integer using n bits.

The *watermarking_enabled_flag* is used to flag whether watermarking will be used or not. Therefore, a value of 0

indicates that the proposed system will not be used and normal H.264/AVC encoding is executed. On the other hand, if the *watermarking_enabled_flag* is set to 1 it indicates that the system proposed in this paper will be used and additional parameters required by the decoder to perform reversible watermark extraction are included.

IV. SIMULATION RESULTS

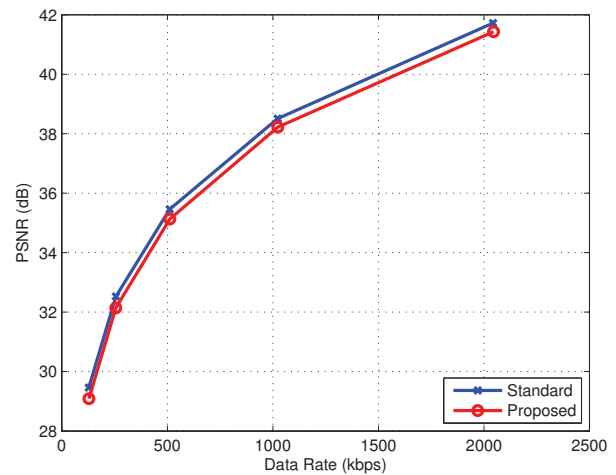
The proposed method was integrated within the joint model (JM) software version 18.3. Both the standard H.264/AVC codec and the proposed scheme were configured to adopt a slice structuring with a fixed number of 200-bytes per slice. An Intra refresh rate of 20% was used to increase the resilience to transmission errors. The encoder was configured to use the IPPP ... format with a GOP size of 15. The *Foreman* and *Coastguard* video sequences at Common Interframe Format (CIF) resolution and 30 fps were considered for the following simulations, which were stored using the 4:2:0 YUV format.

The NALUs encoded using the standard H.264/AVC codec were encapsulated within RTP/UDP/IP, while those encoded using the proposed system were encapsulated within RTP/UDP Lite /IP packets. In both cases, the single NALU packetization mode is used and the IP packets are simulated over a Binary Symmetric Channel (BSC). A total of 50 different noise patterns for each bit error rate (BER) were considered.

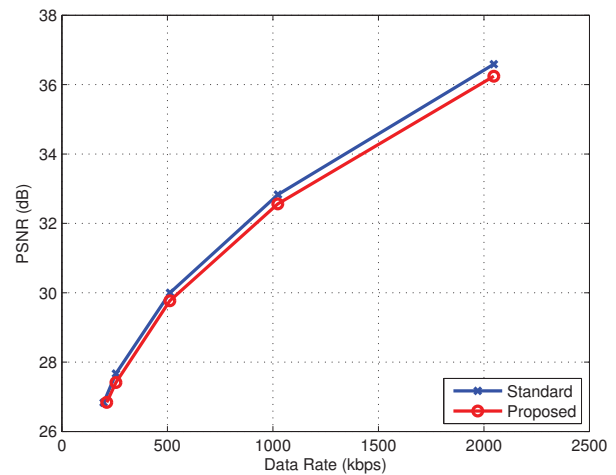
Fig. 4 illustrates the rate distortion curves provided by the encoder. It can be seen that the performance of the proposed scheme achieves lower performance relative to the standard. This is mainly attributed to the fact that the embedded information modifies the image statistics which result in a reduction in compression efficiency. The increase in bitrate at the same PSNR is around 6 ~ 8%, which is similar to the bitrate increase when using standard error resilience tools such as Flexible Macroblock Ordering (FMO) [13]. Moreover, the method presented in [12] had a similar increase in bitrate, without including the bitrate required to compress the location map.

The performance of the proposed scheme when transmitted over error prone channels can be seen in Fig. 5. It can be seen that the proposed scheme outperforms the standard especially at moderate to high error rates. In fact, if a PSNR of 25 dB is considered to indicate a satisfactory level of quality for the proposed system, the standard only achieved a PSNR of 22.5 dB for the *Foreman* and 14.9 dB for the *Coastguard* at the same bit error rate. This can be confirmed from the subjective results illustrated in Fig. 6.

These results demonstrate that in wireless scenarios it is more advantageous to use UDP Lite at the transport layer to protect up to the slice header, and pass the macroblocks contained within a corrupted slice to the decoder. By employing reversible watermarking it is possible to detect, with accuracy higher than 95%, the actual macroblocks that need to be concealed. Moreover, this gain in performance is achieved with a small increase of 0.4 % of computational complexity at both encoder and decoder, which can be considered as negligible.



(a)



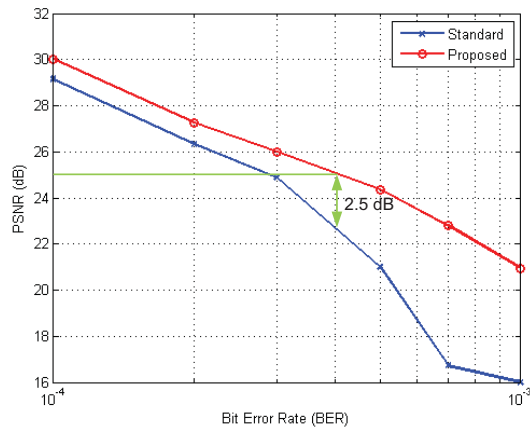
(b)

Fig. 4. Rate distortion curve for the (a) *Foreman* sequence and (b) *Coastguard* sequence.

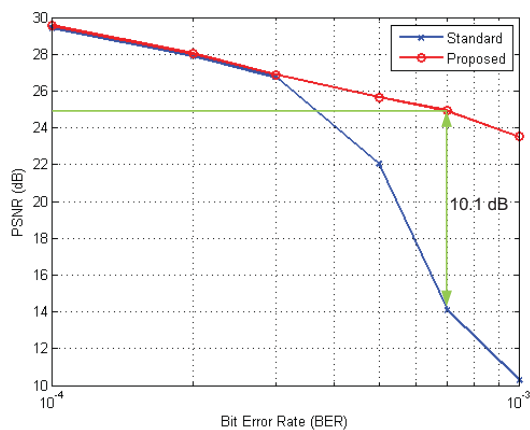
V. COMMENTS AND CONCLUSION

This paper has presented an algorithm which employs reversible watermarking to embed a variable length checksum within the motion vectors and residual information. The decoder processes each macroblock individually, extracts the reversible checksum embedded within it and recomputes the checksum based on the recovered motion vectors and residual information. The macroblocks whose checksums do not match are considered to be corrupted and concealed. The UDP Lite transport layer protocol is used to enable the decoder to receive partially corrupted slices.

The major contribution of this paper is that the decoder manages to recover the original values after watermark extraction. Moreover, this is achieved without requiring the transmission of side information, which was one of the major limitations of previous systems [12]. Simulation results indicate that significant gain in performance can be achieved when simulating the proposed system over a BSC channel. PSNR gains of up to 10.1 dB were registered relative to



(a)



(b)

Fig. 5. Error resilience performance using the (a) Foreman sequence at 512kbps and (b) Coastguard sequence at 1Mbps.

the standard. This significant gain in performance is achieved without significantly affecting the complexity of both encoder and decoder.

REFERENCES

- [1] Blazej Lewcio, Sebastian Möller, Pablo Vidales, Marcel Wältermann, and Niklas Kirschnick, "Methods for Multimedia Service Adaptation in Next Generation Networks," in *Proceedings of the ETSI Workshop on the Effects of Transmission Performance on Multimedia QoS*, Sophia Antipolis, France, June 2008, European Telecommunications Standards Institute (ETSI).
- [2] T. Stockhammer and M. M. Hannuksela, "H.264/AVC video for wireless transmission," *IEEE Wireless Communications*, vol. 12, no. 4, pp. 6–13, Aug. 2005.
- [3] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 657–673, July 2003.
- [4] Lars Larzon, Mikael Degermark, and Stephen Pink, "UDP Lite for Real-Time Multimedia Applications," in *Proceedings of the IEEE International Conference of Communications (ICC)*, 1999.
- [5] L. Superiori, O. Nemethova, and M. Rupp, "Performance of a h.264/AVC error detection algorithm based on syntax check," *Journal of Mobile Multimedia*, vol. 3, no. 4, pp. 314–330, 2007.
- [6] Hang Nguyen and Pierre Duhamel, "Robust source decoding of variable-length encoded video data taking into account source constraints," *IEEE Transactions on Communications*, vol. 53, no. 7, pp. 1077–1084, 2005.



(a)



(b)

Fig. 6. Subjective evaluation for the Foreman sequence at 512kbps (a) Standard H.264/AVC Decoder and (b) Proposed system.

- [7] R.A. Farrugia and C.J. Debono, "Robust decoder-based error control strategy for recovery of h.264/avc video content," *Communications, IET*, vol. 5, no. 13, pp. 1928–1938, 5 2011.
- [8] Ekram Khan, Stefan Lehmann, Hiroshi Gunji, and Mohammed Ghanbari, "Iterative error detection and correction of h.263 coded video for wireless networks," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 14, no. 12, pp. 1294–1307, 2004.
- [9] R.A. Farrugia and C.J. Debono, "A robust error detection mechanism for h.264/avc coded video sequences based on support vector machines," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 12, pp. 1766–1770, dec. 2008.
- [10] O. Nemethova, G.C. Forte, and M. Rupp, "Robust error detection for h.264/AVC using relation based fragile watermarking," in *Proceedings of International Conference on Systems, Signals and Image Processing (IWSSIP)*, Budapest, Ungarn, September 2006.
- [11] Minghua Chen, Yun He, and R. L. Lagendijk, "A fragile watermark error detection scheme for wireless video communications," *Trans. Multi.*, vol. 7, no. 2, pp. 201–211, Apr. 2005.
- [12] R. Facciolo and R.A. Farrugia, "Robust video transmission using reversible watermarking techniques," in *Multimedia (ISM), 2010 IEEE International Symposium on*, dec. 2010, pp. 161–166.
- [13] Farrugia R.A., *Resilient Wireless Transmission of H.264/AVC through Error Localisation and Control Mechanisms*, PhD in Communications engineering, University of Malta, 2008.
- [14] Roberto Caldelli, Francesco Filippini, and Rudy Becarelli, "Reversible Watermarking Techniques: An Overview and a Classification," *EURASIP Journal on Information Security*, vol. 2010, 2010.
- [15] Jun Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, Aug. 2003.
- [16] Hyoung J. Kim, Vasilij Sachnev, Yun Q. Shi, Jeho Nam, and Hyon-Gon Choo, "A Novel Difference Expansion Transform for Reversible Data Embedding," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 456–465, Sept. 2008.
- [17] Dinu Coltuc and Jean-Marc Chassery, "Very fast watermarking by reversible contrast mapping," *CoRR*, vol. abs/0707.0802, 2007.
- [18] Xiao Zeng, Zhenyong Chen, Ming Chen, and Zhang Xiong, "Reversible video watermarking using motion estimation and prediction error expansion," *J. Inf. Sci. Eng.*, vol. 27, no. 2, pp. 465–479, 2011.