A Robust Error Detection Mechanism for H.264/AVC Coded Video Sequences Based on Support Vector Machines

Reuben A. Farrugia, *Member, IEEE*, and Carl James Debono, *Senior Member, IEEE*

***Abstract*—Current trends in wireless communications provide fast and location-independent access to multimedia services. Due to its high compression efficiency, H.264/AVC is expected to become the dominant underlying technology in the delivery of future wireless video applications. The error resilient mechanisms adopted by this standard alleviate the problem of spatio-temporal propagation of visual artifacts caused by transmission errors by dropping and concealing all macroblocks (MBs) contained within corrupted segments, including uncorrupted MBs. Concealing these uncorrupted MBs generally causes a reduction in quality of the reconstructed video sequence.**

This paper presents a novel error detection algorithm which employs the checksum of the transport layer protocol to detect corrupted segments. Each MB within the corrupted segment is passed through a support vector machine (SVM) classifier to detect and localize visually distorted MBs. The proposed solution was tested on a wide range of video sequences, where on average 95.25% of the residual corrupted MBs which provide annoying visual artifacts were detected. This method reduces the number of uncorrupted MBs to be concealed resulting in a significant gain in quality compared to the standard H.264/AVC decoder.

***Index Terms*—Error-resilient video coding, H.264/ AVC, learning systems, video coding, wireless video transmission.**

I. Introduction

MULTIMEDIA communication services have become part of everyday life where location-independent access to these services has turned out to be a requirement. H.264/AVC is an attractive candidate for all wireless video applications mainly because of its enhanced compression efficiency and network friendly design [1]. However, H.264/AVC is susceptible to transmission errors where even a single corrupted bit may cause visual artifacts that may propagate in the spatio-temporal domain. This problem mainly affects conversational and multicast/broadcast applications due to delay constraints and the unavailability of a feedback channel respectively [2]. For this reason, these applications generally adopt slice level concealment (SLC) where the checksum of the transport protocol is used to detect corrupted slices. All macroblock (MBs) contained within corrupted slices are dropped and concealed. However, each corrupted slice can contain a large number of uncorrupted MBs which are also concealed, thus reducing the quality of the reconstructed video sequence. A model which tries to predict the video quality in terms of visibility of impairment in similar environments is given in [3].

The authors in [4] have adopted syntax and semantic tests to detect invalid codewords and apply MB level concealment (MLC) for H.264/AVC coded sequences. However, this method only manages to detect 57% of the corrupted MBs. In [5]–[8] corrupted MBs were detected in the pixel domain using heuristic thresholds. These methods have limited applications since the optimal thresholds vary from sequence to sequence. In [9] an iterative solution was used to recover corrupted MBs. This method however is computational intensive and thus cannot be applied in real-time applications.

Data hiding techniques were adopted in [10]–[12], however these methods require additional complexity in both the encoder and decoder. The method in [13] enhances the error resilience by applying parity bits and synchronization markers which reduce the compression efficiency of the codec. The author in [14] applies source constraints to detect corrupted motion information in MPEG-4 video sequences, while Sequential [15], [16] and List [17] decoders were adopted to recover the most-likelihood bitstreams. However, these methods do not distinguish between annoying and imperceptible artifacts thus wasting resources.

To the knowledge of the authors, classification methods were only used to detect visually distorted MBs in [18] where a Probabilistic Neural Network was adopted in H.263 encoded sequences at CIF resolution. This paper presents an algorithm which is optimized to detect corrupted MBs which provide annoying visual artifacts within H.264/AVC video sequences, and is targeted for video-telephony and multicast-broadcast applications. Corrupted segments, which contain a set of potentially corrupted MBs, are detected using the checksum of the transport protocol. Once an MB is flagged as potentially corrupted a set of image-level features are extracted. These are fed to a support vector machine (SVM) which detects the visually impaired MBs to be concealed. Results show that this algorithm manages to detect 95.25% of the visually impaired MBs, hence providing a significant gain in quality.

This paper is organized as follows. The proposed error detection algorithm is described in Section II where details about the added components are given. Section III presents the simulation results which highlight the gain in both subjective and objective quality. The final comments and conclusion are presented in Section IV.

II. Proposed Error Detection Algorithm

The proposed error detection technique is incorporated within the modified decoder. All protocol stacks used to deliver real-time video content such as H.324M and RTP/UDP/IP employ bit-level error checksums to detect corrupted packets [1]. This

Manuscript received July 19, 2008; revised November 01, 2007 and January 23, 2008. First published September 19, 2008; current version published November 26, 2008. This paper was recommended by Associate Editor T. Nguyen.

The authors are with the Department of Communications and Computer Engineering, University of Malta, Msida, MSD 2080, Malta (e-mail: rrfarr@eng.um.edu.mt; cjdebo@eng.um.edu.mt).

Digital Object Identifier 10.1109/TCSVT.2008.2004919

property can be used to flag as potentially corrupted the MBs encapsulated within corrupted packets. The proposed algorithm is then only applied on these flagged MBs, thus no extra processing is incurred when uncorrupted packets are received.

A set of image-level features are then extracted for each potentially corrupted MB. These features exploit the inherent redundancies within the MB, peripheral MBs and with temporally corresponding MBs. The resulting features are passed through an SVM classifier which detects the corrupted MBs that provide annoying visual distortions. These MBs are concealed using the weighted-pixel value averaging for INTRA pictures [19] and boundary-matching based motion vector recovery for INTER pictures [20] which are used in the Joint Model (JM) reference [24] implementation of the H.264/AVC standard.

A. Feature Extraction

The visual artifacts caused by transmission errors present in the channel generate different levels of distorted MBs being perceived by the end-user; some of them are very annoying while others imperceptible. In this work it was found that most of the corrupted MBs provide imperceptible visual artifacts (77.48% for *Foreman* sequence with a BER of 1.00E-003). Therefore, concealing all the corrupted segments generally results in concealing a number of uncorrupted MBs, causing an unnecessary reduction in the perceptual quality. For this reason, image-level features were adopted.

A large set of features was chosen which on our experience and judgment would help to detect different visual artifacts. However, since the complexity of the SVM is dependent on the number of features being used, different combinations of features were simulated and tested. The best performance registered by the SVM classifier was achieved when the following set of pixel domain features was adopted; the Average Inter-sample Difference across Boundaries (AIDB), 16 Internal AIDB (IAIDB) features, and 16 Average Internal Difference between Subsequent Blocks (AIDSB) features. These features were computed in the perceptually uniform CIELUV color space model, since these metrics were found to perform best in this color space [6]. This is mainly because the distance in the CIELUV color space generally corresponds to the difference in color sensation.

Average Inter-Sample Difference Across Boundaries: In an image, there exists sufficient similarity among the adjacent pixels, and hence across MB boundaries, although there may exist clear edges along their boundaries. The AIDB feature detects MBs whose distortion affects the entire 16 × 16 MB. From Fig. 1, let M represent a potentially corrupted MB under test with its four neighboring MBs; N, S, E, and W. Further, let $p^{\text{in}} = \{p_1^{\text{in}}, p_2^{\text{in}}, \ldots, p_K^{\text{in}}\}$ represent boundary pixels inside the MB M, and $p^{\text{out}} = \{p_1^{\text{out}}, p_2^{\text{out}}, \ldots, p_K^{\text{out}}\}$ represent boundary pixels in one of the neighboring MBs. Then, the AIDB of M with an MB $X \in \{N, S, E, W\}$, $AIDB(M:X)$ is:

$$AIDB(M:X) = \begin{cases} \frac{1}{K}\|p^{\text{in}} - p^{\text{out}}\|_2, & \text{if X available} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where K is the size of the MB ($K = 16$) and $\| \bullet \|_2$ is the L^2 norm which is computed in the CIELUV color space model.

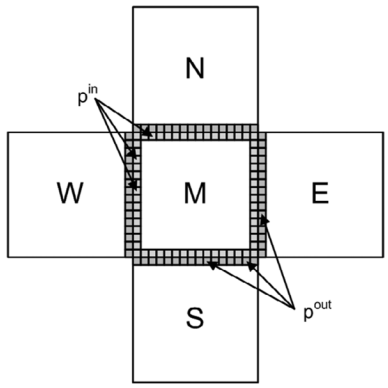

Fig. 1. Boundary pixels for a KxK block.

The $AIDB$ feature is then computed by evaluating the average of $AIDB(M:X)$ over the available neighboring MBs.

Internal AIDB: A corrupted MB does not always result in all the contained pixels being distorted. Since the H.264/AVC design is based primarily on 4 × 4 transforms, the MB is divided in a grid of 16 4 × 4 blocks. To measure the similarity among boundary pixels across boundaries within an MB, the $IAIDB$ features were defined. These features are similar to the $AIDB$ defined earlier, with the difference that $IAIDB$ is defined within an MB.

The $IAIDB(M:X)$ for each block is computed using (1), where $X \in \{N, S, E, W\}$ corresponds to the neighboring 4 × 4 blocks. The $IAIDB$ feature for each 4 × 4 block is then derived by averaging the $IAIDB(M:X)$ for each 4 × 4 block.

Average Internal Difference Between Subsequent Blocks: Generally, the pixel transition of an MB and the corresponding MB in the previous frame varies smoothly. Again, since the H.264/AVC design is based on 4 × 4 transform blocks, each MB is dissected into 16 4 × 4 blocks. Let M_t represent the potentially corrupted MB under test and M_{t-1} be the corresponding MB in the previous frame, then the $AIDSB$ feature for each 4 × 4 block is computed using

$$AIDSB = \frac{1}{K^2}\|p_t - p_{t-1}\|_2 \quad (2)$$

where K represents the size of the block (in this case $K = 4$), $\| \bullet \|_2$ is the L^2 norm computed in CIELUV color space, and p_t and p_{t-1} represent the pixel of the 4 × 4 block under test and the corresponding block in the previous MB respectively.

B. Support Vector Machine (SVM) Classifier

In supervised learning, the learning machine is given a training set S which is denoted by

$$S = [(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_l, y_l)], \vec{x} \in \Re^N, y \in \{-1, +1\} \quad (3)$$

where $\vec{x}$ are the extracted feature vectors, y their labels (corrupted MB $= -1$, uncorrupted MB $= +1$), l is the number of examples, and N represent the number of dimensions of the feature vector. In order to discover the nonlinear relations within a linear machine, the SVM [21] employs an

TABLE I
DEFINITION OF THE DISTORTION LEVELS

Distortion Level (DL)	Definition
4	Very annoying artifact
3	Annoying artifact
2	Slightly annoying artifact
1	Perceptible but not annoying artifact
0	Imperceptible or uncorrupted MB

implicit nonlinear mapping of the data onto a higher dimensional feature space via a positive semi-definite kernel $K(x, y)$, where the SVM tries to derive an optimized hyperplane.

The SVM was trained using a modified version of the Sequential Minimal Optimization (SMO) [22] to solve the following quadratic optimization problem:

$$\max_{\alpha} \; W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j K(\vec{x}_i, \vec{x}_j) \alpha_i \alpha_j$$
$$\text{s.t.} \sum_{i=1}^{l} y_i \alpha_i = 0, \; 0 \le \alpha_i \le C, \; \forall i \tag{4}$$

where α are the Lagrange multipliers, and C is the finite penalization constant. The Lagrange multipliers have nonzero values to support vectors (SV), which solely determine the optimal hyperplane. The feature vector describing the MB is assigned to a class based on the following rule:

$$f(x) = \text{sign}\left(\sum_{i \in sv} \alpha_i y_i K(\vec{x}_i, \vec{x}) + b \right) \tag{5}$$

where b is the bias. The SVM classifier adopted utilizes a Gaussian Kernel which is given by:

$$K(x, y) = \exp\left(\frac{-||x - y||^2}{2\sigma^2} \right) \tag{6}$$

where σ is the smoothing parameter. Following extensive testing the penalization constant C was set to 60 while the smoothing factor σ was set to 4.325.

The training and the testing set supplied to the SVM for the training and the recognition phases consisted of 1000 feature vectors each (500 uncorrupted MBs and 500 corrupted MBs). These MBs were selected at random from a set of five video sequences at QCIF resolution: *Foreman, Car-phone, Mobile, Coastguard*, and *News*. Four other video sequences *Miss-America, Container, Salesman* and *Akiyo*, were adopted for cross-validation. All these sequences were encoded with both FMO switched on and off and at a BER of 1.00E-004.

The subjective categorization of the MBs used during testing and cross-validation were scaled using a methodology similar to the single stimulus test [23] and adopted in [3]. The 18 viewers used in the subjective tests had to indicate which MB is corrupted within a video sequence and to scale each artifact according to the five-level distortion scale given in Table I. The Mean Opinion Score (MOS) was then used to provide a subjective rating of the distortion level of the MBs.

TABLE II
ERROR DETECTION RATES OF DISTORTED MBS

Distortion Level (DL)	Error Detection Rate Recognition	Error Detection Rate Cross-Validation
Very Annoying Artifact (4)	100.00%	100.00%
Annoying Artifact (3)	100.00%	100.00%
Slightly Annoying Artifact (2)	92.16%	86.46%
Perceptible but not Annoying (1)	61.00%	64.77%
Overall	91.60%	89.24%

III. SIMULATION RESULTS

The aim of the proposed solution was to maximize the detection of highly distorted MBs which significantly distort the quality of the decoded frame while being more lenient with imperceptible ones. Table II summarizes the recognition and cross-validation performance provided by the SVM on the testing set given different distortion levels (DL), which were scaled according to a survey as described in Section II-B.

These results show that on average the SVM has managed to detect 95.25% of the corrupted MBs which provide major visual distortion (DL4, DL3, and DL2) together with 62.91% of the DL1 artifacts. Since the undetected MBs provide minor or almost imperceptible visual artifacts the quality of the reconstructed video sequence will not be significantly affected. Moreover, the false-positives which are generated are only 5.20%, and are bounded within the set of potentially corrupted MBs, which at the worst will function like the SLC method. The proposed error detection algorithm was integrated within the jointed model (JM) software [24], together with the syntax check rules described in [4]. The raw video was encoded at QCIF resolution at 30 fps, with the format IPPP... and a data rate of 128 kbps. The encoder applies a random intra-refresh rate of 5 and has five slices per picture with FMO switched on and 1 slice per picture with FMO switched off. In compliance with JM software, each slice was encapsulated within RTP/UDP/IP packets, thus ensuring synchronization at slice level. The UDP transport protocol was modified to protect the header information in a similar way to the UDP Lite protocol [25]. If a corrupted UDP header is detected, the slices contained within the packet are dropped and concealed. Otherwise, the corrupted slices are handled by the proposed algorithm at the application layer. To validate the proposed solution the compressed bitstream was transmitted over two different channels: 1) binary symmetric channel (BSC) which yields the highest probability of error in the slice [9] and 2) the DVB-RCS channel adopted in many digital video broadcasting applications [26] modeled as in [27], which is an example of a bursty channel. The sequences were repeated with ten different starting positions in the error pattern to obtain more reliable results.

The received corrupted bitstream was decoded using three different error detection algorithms: 1) MLC, where whenever a syntax rule is violated the current MB and the following ones are concealed until the end of slice; 2) SLC, where every time an error is detected by the UDP error checksum the entire slice is discarded and concealed; and 3) the proposed algorithm (SVM), which conceals only those MBs which according to the SVM

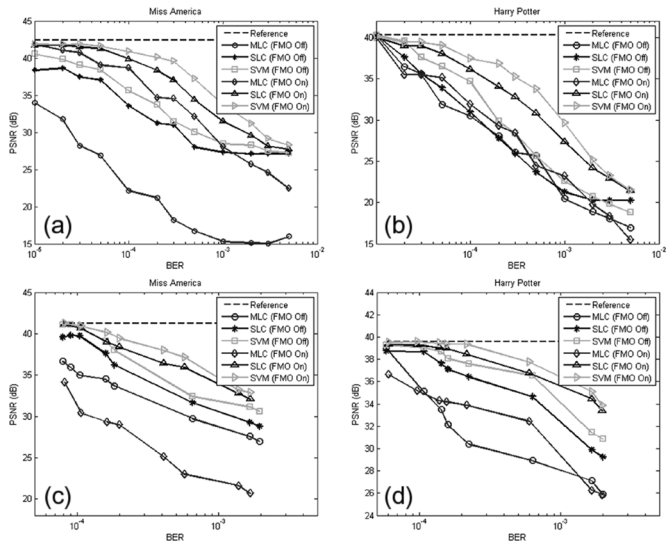

Fig. 2. PSNR gains of the three considered error detection algorithms on (a) Miss-America (BSC Channel), (b) Harry Potter (BSC Channel), (c) Miss-America (DVB-RCS Channel), and (d) Harry Potter (DVB-RCS Channel) sequences at different BER.

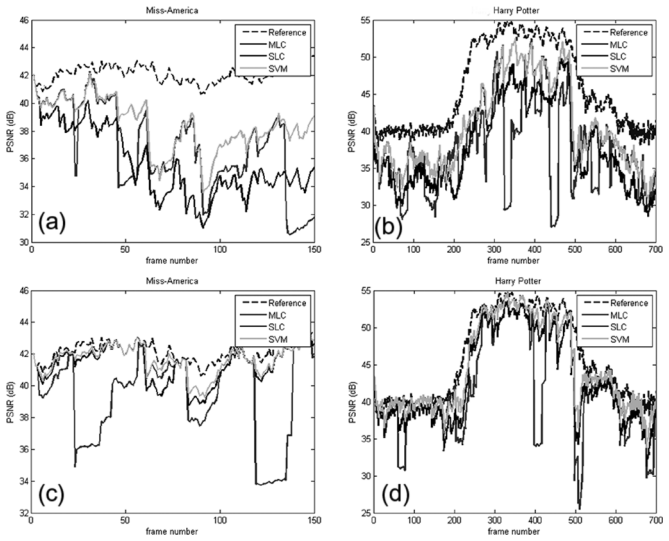

Fig. 3. PSNR gains of the three considered error detection algorithms on (a) Miss-America (BSC Channel), (b) Harry Potter (BSC Channel), (c) Miss-America (DVB-RCS Channel), and (d) Harry Potter (DVB-RCS Channel) sequences at a BER of 1.00E-004.

classifier are corrupted. The performance of these algorithms at different BER is illustrated in Fig. 2. The video sequences that are considered during testing are: 1) *Miss-America* which is a typical videoconferencing application and 2) 8900 frames from the movie *Harry Potter (The Goblet of Fire)* which represent a typical broadcasting application. These sequences were not used in the training phase and thus the results are not biased.

From the figures above, it is clear that the proposed algorithm outperforms the MLC and SLC methods. Moreover, additional error resilience can be achieved when dispersed FMO is enabled. The performance of the proposed algorithm relative to MLC and SLC on a frame by frame basis is illustrated in Fig. 3 at a BER of 1.00E-004. This time the *Miss-America* sequence was encoded with FMO switched on with five slices per picture while the *Harry Potter* sequence was encoded with FMO switched off with one slice per picture. The proposed algorithm achieves PSNR gains of up to 8.36 and 6.14 dB relative to MLC and SLC methods when FMO is switched on, and 20.10 and 6.65 dB when FMO is switched off.

These results confirm the superiority of the proposed algorithm. This is mainly because the MLC algorithm does not manage to detect a number of corrupted MBs, some of which provide major visual distortions that will be propagated in both the spatial and the temporal domains. On the other hand, the SLC algorithm is pessimistic, and although it detects most of the errors at slice level, it conceals the whole slice even if only one MB is corrupted.

A small number of MBs remain still undetected by the SVM, however these generally provide DL1 or DL2 artifacts which are generally tolerated by the end-users. On the other hand, DL3 and DL4 artifacts are generally detected by the SVM solution, as shown in Table II above. The gain in subjective quality provided by the proposed solution is illustrated in Fig. 4 where different frames of the *Harry Potter* sequence are illustrated.

Fig. 4. Harry Potter sequence using (left) MLC, (center) SLC, and (right) proposed method.

IV. Comments and Conclusion

This paper has presented a novel solution which can be used to enhance the error detection capabilities of the standard H.264/AVC decoder. The proposed method manages to detect most of the corrupted MBs which provide major distortions while being more lenient on MBs which provide unnoticeable visual artifacts. This reduces the number of MBs that are concealed resulting in significant gains in both objective and subjective quality. The main limitation of the proposed solution is that the transport layer was modified. However, flexible transport protocols, such as UDP Lite [25], allow the delivery of partially damaged payloads.

Furthermore, the proposed algorithm comes only into action whenever a segment is detected to be corrupted by the transport protocol. This implies that the added decoding time increases with increasing BER, where an 8.87% increase in complexity was experienced at a BER of 5.00E-003. This result

shows that the complexity added at the decoding side is minimal, thus making the proposed system usable in real-time applications such as video telephone and multicast/broadcast applications. Future work will address the application of this algorithm with temporally scalable coding patterns which are often used in multicast/broadcast applications.

REFERENCES

[1] T. Stockhammer, M. N. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 657–673, Jul. 2003.

[2] T. Stockhammer and M. M. Hannuksela, "H.264/AVC video for wireless transmission," *IEEE Wireless Commun.*, vol. 12, no. 4, pp. 6–13, Aug. 2005.

[3] A. R. Reibman and D. Poole, "Predicting packet-loss visibility using scene characteristics," in *Proc. IEEE Packet Video*, Lausanne, Switzerland, Nov. 2007.

[4] L. Superiori, O. Nemethova, and M. Rupp, "Performance of a H.264/AVC error detection algorithm based on syntax analysis," in *Proc. 4th Int. Conf. Advances Mobile Computing Multimedia*, Yogyakarta, Indonesia, 2006, pp. 1–10.

[5] L. Superiori, O. Nemethova, and M. Rupp, "Detection of visual impairments in the pixel domain of corrupted H.264/AVC packets," presented at the IEEE Int. Picture Coding Symp., Lisbon, Nov. 2007.

[6] R. A. Farrugia and C. J. Debono, "Enhancing the error detection capabilities of the standard video decoder using pixel domain dissimilarity metrics," in *Proc. IEEE EUROCON*, Warsaw, Poland, 2007, pp. 1085–1090.

[7] R. A. Farrugia and C. J. Debono, "Enhancing the error detection capabilities of DCT based codecs using compressed domain dissimilarity metrics," in *Proc. IEEE Int. Conf. EUROCON*, Warsaw, Poland, Sept. 2007, pp. 1091–1095.

[8] S. Ye, X. Lin, and Q. Sun, "Content based error detection and concealment for image transmission over wireless channel," in *Proc. IEEE Int. Symp. Circuits Systems*, Bangkok, Thailand, 2003, pp. 368–371.

[9] E. Khan, S. Lehmann, H. Gunji, and M. Chanbari, "Iterative error detection and correction of H.263 coded video for wireless networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 12, pp. 1294–1307, Dec. 2004.

[10] O. Nemethova, G. C. Forte, and M. Rupp, "Robust error detection for H.264/AVC using relation based fragile watermarking," presented at the IEEE 13th Int. Conf. Systems Image Processing, Budapest, Hungary, 2006.

[11] M. Chen, Y. He, and R. L. Lagendijk, "A fragile watermarking error detection scheme for wireless video communications," *IEEE Trans. Multimedia*, vol. 7, no. 2, pp. 201–211, Apr. 2005.

[12] C. B. Adsumilli, M. C. Q. Farias, S. K. Mitra, and M. Carli, "A robust error concealment technique using data hiding for image and video transmission over lossy channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 11, pp. 1394–1406, Nov. 2005.

[13] O. Nemethova, J. C. Rodriquez, and M. Rupp, "Improved detection for H.264 encoded video sequences over mobile networks," presented at the IEEE 8th Int. Symp. Commun. Theory Appl., Ambleside, Lake District, U.K., 2005.

[14] B. Yan and K. W. Ng, "Analysis and detection of MPEG-4 visual transmission errors over error-prone channels," *IEEE Trans. Consum. Electron.*, vol. 49, no. 4, pp. 1424–1430, Nov. 2003.

[15] C. Weidmann, P. Kadlec, O. Nemethova, and A. Al Moghrabi, "Combined sequential decoding and error concealment of H.264 video," in *Proc. Int. IEEE 6th Workshop Multimedia Signal Process.*, Siena, Italy, Oct. 2004, pp. 299–302.

[16] C. Bergeron and C. Lamy-Bergot, "Soft-input decoding of variable-length codes applied to the H.264 standard," in *Proc. Int. IEEE 6th Workshop Multimedia Signal Process.*, Siena, Italy, Oct. 2004, pp. 87–90.

[17] H. Nguyen, P. Duhamel, J. Brouue, and D. Rouffet, "Optimal VLC sequence decoding exploiting additional video stream propoerties," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Montreal, QC, Canada, May 2004, pp. 621–624.

[18] R. A. Farrugia and C. J. Debono, "Enhancing error resilience in wireless transmitted compressed video sequences through a probabilistic neural network core," presented at the IEEE Int. Picture Coding Symp., Lisbon, Portugal, 2007.

[19] P. Salama, N. B. Shroff, and E. J. Delp, "Error concealment in encoded video," in *Image Recovery Techniques for Image Compression Applications*. Norwell, MA: Kluwer, 1993.

[20] W. M. Lam, A. R. Reibman, and B. Liu, "Recovery of lost erroneously received motion vectors," in *Proc. ICASSP*, Mar. 1993, vol. 5, pp. 417–420.

[21] N. Cristianini and J. D. Taylor, *Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: CUP, 2000.

[22] S. S. Keerth, S. K. Shevade, C. Bhattacharyyam, and K. R. K. Murthy, "Improvements to Platt's SMO algorithm for SVM classifier design," *Neural Comp.*, vol. 13, no. 3, pp. 637–649, Mar. 2001.

[23] *Subjective Video Quality Assessment Methods for Multimedia Applications* ITU-T REc., p. 910, 1999.

[24] JM Software ver. 12.2, H.264/AVC Software Coordination [Online]. Available: http://iphone.hhi.de/suehring/tml

[25] L.-A. Larzon, M. Degermark, S. Pink, L.-E. Jonsson, and G. Fairhurst, "The Liteweight User Datagram Protocol (UDP-Lite)," RFC-3282, Jul. 2004.

[26] ETSI EN 301 790, *Digital Video Broadcasting (DVB); Interactive Channel for Satellite Distribution systems* 2005.

[27] R. A. Farrugia and C. J. Debono, "A statistical bit error generator for emulation of complex forward error correction schemes," in *Proc. Int. Conf. Communications (ICC)*, Glasgow, Scotland, Jun. 2007, pp. 177–182.