Improved Quality of Experience of Reconstructed H.264/AVC Encoded Video Sequences through Robust Pixel Domain Error Detection

Reuben A. Farrugia, Carl J. Debono

Department of Communications and Computer Engineering, University of Malta
Msida, MSD2080, Malta

reuben.farrugia@um.edu.mt
c.debono@ieee.org

***Abstract*—The transmission of H.264/AVC encoded sequences over noisy wireless channels generally adopt the error detection capabilities of the transport protocol to identify and discard corrupted slices. All the macroblocks (MBs) within each corrupted slice are then concealed.**

This paper presents an algorithm that does not discard the corrupted slices but tries to detect those MBs which provide major visual artefacts and then conceal only these MBs. Results show that the proposed solution, based on a set of image-level features and two Support Vector Machines (SVMs), manages to detect 94.6% of those artefacts. Gains in Peak Signal-to-Noise Ratios (PSNR) of up to 5.74 dB have been obtained when compared to the standard H.264/AVC decoder.

I. Introduction

The demand for real-time and location-independent multimedia services is experiencing a steady increase requesting larger data volumes to be transmitted within the limited bandwidths available. This, together with restrictions in transmission power, makes compression efficiency a major concern in wireless multimedia applications [1].

H.264/AVC is a promising solution and is expected to become an essential component in emerging wireless applications due to its excellent compression efficiency and network friendly design [2]. However, like all other video compression standards, H.264/AVC is extremely vulnerable to transmission errors. This results in annoying visual artefacts which will propagate in both the spatial and temporal domains, significantly degrading the perceptual quality of the decoded video sequences.

In order to alleviate the spatio-temporal propagation of distorted macroblocks (MBs), the checksum of the transport layer protocols is used to identify corrupted slices. The standard decoder generally applies slice level concealment (SLC) on the erroneous slices and conceals all the MBs contained within each corrupted slice.

Macroblock level concealment (MLC) was considered in [3]. However, this method only manages to detect 57% of the corrupted MBs leaving artefacts which significantly reduce the quality of the recovered video sequences. The method proposed in [4] utilizes synchronization markers and parity bits to enhance the error resilience of H.264/AVC coded bitstreams. Yet, this is achieved at the cost of reducing the compression efficiency of the codec. In a different approach, the authors in [5] applied watermarking for the detection of corrupted MBs. This method however introduces additional complexities in both the encoder and decoder. The authors in [6], [7] have adopted Sequential Decoding in order to recover the most-likelihood bitstream. Although these solutions provide additional robustness to the transmitted bitstream, they do not discriminate between annoying and imperceptible artefacts, and thus resources are wasted in correcting unnoticeable artefacts. In [8] the authors have detected corrupted MBs in the pixel domain using heuristic thresholds but this method requires a change in threshold for each video sequence. In [9] and [10] a Probabilistic Neural Network (PNN) was adopted to detect corrupted MBs at image level for H.263 and H.264/AVC encoded sequences, respectively.

In this paper we present a robust error-resilient mechanism which applies the checksum present in the transport protocol to detect corrupted slices. A set of image-level features in conjunction with a pair of Support Vector Machines (SVMs) are then applied to the corrupted slices to identify the visually distorted MBs to be concealed. This provides a good compromise between the SLC and MLC methods with Peak Signal-to-Noise (PSNR) gains of 14.46 dB and 5.74 dB being observed when compared to the MLC and the standard decoder respectively.

This paper is organized as follows: The proposed error detection algorithm is presented in section II together with the detail on the components that need to be added to the standard decoder. Section III discusses the support vector machine followed by simulation results highlighting the gain in both subjective and objective quality in section IV. The final conclusions are presented in Section V.

II. Proposed Error Detection Algorithm

The proposed error detection algorithm is incorporated within the modified standard H.264/AVC decoder, as shown in Fig. 1. The protocols adopted to deliver real-time multimedia content, such as RTP/UDP/IP, employ bit-level checksums to detect any corrupted segments [1]. The checksum of the User Datagram Protocol (UDP) was used in the proposed system to detect the corrupted slices. All the MBs contained within corrupted slices are flagged as being potentially corrupted. The proposed algorithm is applied only on these flagged MBs, thus no additional delay is introduced in the system when the transmitted video content is uncorrupted.

Each potentially corrupted MB passes through the first stage of the algorithm where a set of image level features are extracted. These features exploit the inherent redundancies within the MB, neighbouring MBs and with temporally corresponding MBs. Following this, the set of potentially corrupted MBs is reduced by the first SVM (SVM1) while detection of false positives is suppressed by the second SVM (SVM2).

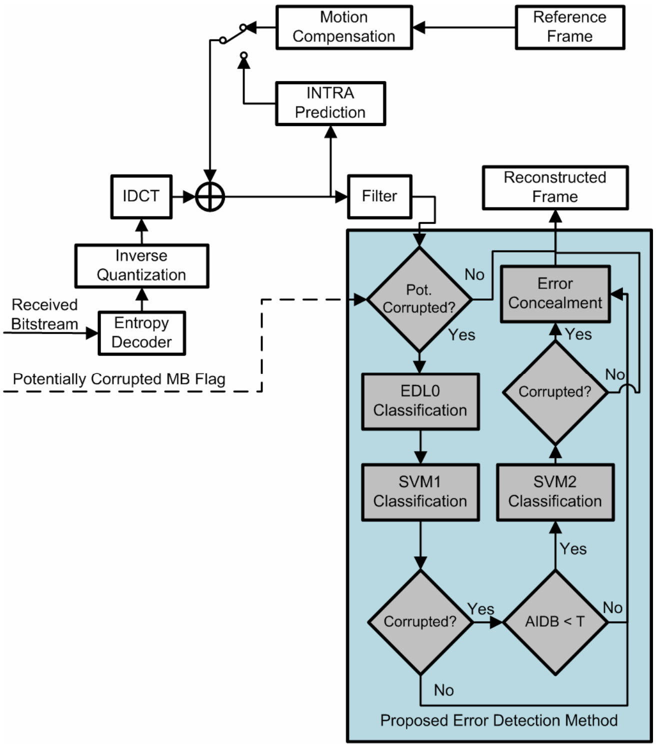


Fig. 1 Proposed Error Detection Algorithm

A set of eight image-level features was found to provide enough information to the classifiers to discriminate between corrupted and uncorrupted MBs. These are:

- *Average Inter-sample Difference across Boundaries (AIDB)* [9] which describes the spatial fitness of an MB with its neighbouring MBs.
- The mean and standard deviation of the *Internal AIDB of the Block (IAIDB$_{Block}$)* [9] which describes the fitness of the internal 4x4 blocks in space with its neighbouring blocks.
- The vertical and horizontal *Internal AIDB (IAIDB)* which describe the consistency across the boundaries of an MB.
- The mean and standard deviation of the *Average Internal Difference between Subsequent Blocks (AIDSB)* [9] which describes the temporal consistency of each 4x4 block.
- *Texture Consistency (TC)* [10] which describes the texture consistency between temporally corresponding MBs.

These features are summarized in the following subsections.

A. Average Inter-sample Difference across Boundaries

In natural video sequences there exists sufficient similarity across MB boundaries. Given a set of neighbouring MBs $X \in \{N, S, E, W\}$ as shown in Fig. 2a, the sum of Euclidean distance across each MB boundary is computed using:

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

where K is the size of the MB and $\|\ \|_2$ is the L^2 norm computed in the CIELUV colour space model. The *AIDB* feature is then computed by evaluating the average *AIDB(M:X)* over the available neighbouring MBs. The larger the value of this feature the less the considered MB fits in space.

B. Internal Average Inter-sample Difference across Boundary Blocks

Each MB is divided in a grid of sixteen 4x4 blocks. The sum of Euclidean distances at each block boundary is computed using (1), where $X \in \{N, S, E, W\}$ now correspond to the neighbouring 4x4 blocks and K = 4. The *IAIDB$_{Block}$* dissimilarity metric is then derived by averaging the *AIDB(M:X)* for each 4x4 block. The mean and standard features are then computed over all sixteen *IAIDB$_{Block}$* dissimilarity metrics.

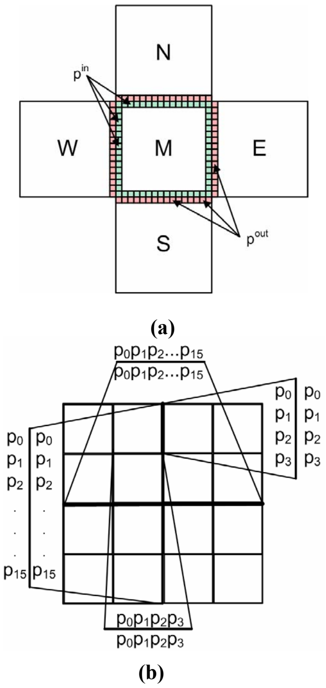


Fig. 2 (a) AIDB (b) IAIDB$_{Block}$ and IAIDB features

This feature describes the fitness of each 4x4 block with its neighbouring blocks. Large values of this feature represent potentially corrupted regions within the considered MB while small values generally correspond to uncorrupted regions.

C. *Internal Average Inter sample Difference across Boundaries*

The *IAIDB* feature is computed by averaging the sum of Euclidean distances between internal vertical and horizontal boundary pixels, as shown in Fig. 2b, using:

$$IAIDB = \sum_{i=0}^{K-1} \left\| p_i^{in} - p_i^{out} \right\|_2 \qquad (2)$$

where K = 16 and $\| \|_2$ is the L^2 norm computed in CIELUV colour space model.

D. *Average Internal Difference between Subsequent Blocks*

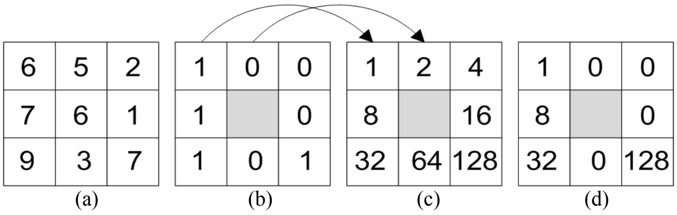
Generally, the pixel transition of an MB and the corresponding MB in the previous frame varies smoothly. Each MB is dissected into sixteen 4x4 blocks and the *AIDSB* dissimilarity metric for each block is computed using:

$$AIDSB = \frac{1}{K^2} \left\| p_t - p_{t-1} \right\|_2 \qquad (3)$$

where p_t and p_{t-1} represent the corresponding pixels within the current and previous MB respectively. The mean and standard deviation are computed over all 16 *AIDSB* metrics. Large values of *AIDSB* generally describe corrupted regions within MBs.

E. *Texture Consistency*

The spatial structure of the local texture of a potentially corrupted MB is computed using the Local Binary Pattern (LBP) on the luminance colour component [11]. The original 3x3 neighbourhood (Fig. 3a) is thresholded by the centre pixel. The values obtained (Fig. 3b) are multiplied by the binomial weights (Fig. 3c) and the resulting values (Fig. 3d) are summed for the LBP texture unit.



(a)

6	5	2
7	6	1
9	3	7

(b)

1	0	0
1		0
1	0	1

(c)

1	2	4
8		16
32	64	128

(d)

1	0	0
8		0
32	0	128

Fig. 3 Computation of the Local Binary Pattern

The LBP histogram for the current h_t and the corresponding MB in the previous frame, h_{t-1}, is then computed. The *TC* feature is derived by using the histogram intersection method;

$$TH = \sum_{i=0}^{B} \min\left(h_t, h_{t-1}\right) \qquad (4)$$

where B is the number of bins. A large value of *TC* signifies textural difference between the two temporally neighbouring MBs and thus a possible distorted MB.

III. Support Vector Machine (SVM)

The SVM [12] is a classifier which implicitly maps the data to a high dimensional feature space via a positive semi-definite kernel *K(x,y)*. The SVM was trained using a modified version of the Sequential Minima Optimization (SMO) [13] algorithm in order to derive the maximal separating hyperplane.

The discrimination between corrupted and uncorrupted MBs is based on the following rule:

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

where α are the Lagrange multipliers, x and y are the training vectors and targets, z is the vector under test and b is the bias. The support vectors (SV) solely determine the optimal hyperplane and therefore their impact on the complexity of the system results only in a minimal increase, making the adaptation of the SVM algorithm applicable in real-time applications like the one considered in this work.

Both SVMs adopted by the proposed system utilize the Gaussian Kernel given by:

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

where σ is the smoothing parameter.

The eight features described in the previous section were applied to SVM1 which manages to detect 95.4% of the corrupted MBs. In order to suppress false detections, the *AIDB* feature is compared to a threshold T, where, after extensive simulations, a value of 0.25 was found to provide the best performance. Only those MBs which have $AIDB < T$ are considered as potential false detections and thus passed to SVM2, while the others are concealed. The eight features are concatenated together with the probabilistic output of SVM1 [14] to form the input feature vector of SVM2 which decides whether the inputted MB is corrupted or not.

IV. Simulation Results

The proposed error detection algorithm was integrated within the Jointed Model (JM) software [15]. The raw video was encoded at QCIF resolution using 5 slices per picture with dispersed Flexible Macroblock Ordering (FMO). Each slice was encapsulated within RTP/UDP/IP packets and transmitted over a binary symmetric channel (BSC). The proposed algorithm was tested on two video sequences: *Miss-America* representing a typical conversational application with minimal movements and *Foreman* a typical application containing camera movements.

The support vector machines need training before they can be applied to a problem. The set of vectors that was supplied to both SVM1 and SVM2 during 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. The overall recognition rate achieved by the proposed system is 94.6%. This recognition rate was registered when testing was performed on a wide range of video sequences, confirming the flexibility of our solution.

Fig. 4 and 5 further summarize the performance of the proposed algorithm when compared to the MLC and the standard SLC techniques. It can be immediately noticed that the proposed algorithm outperforms the other two approaches, with gains in PSNR that go up to 14.46 dB relative to MLC and 5.74 dB when compared to the standard SLC.

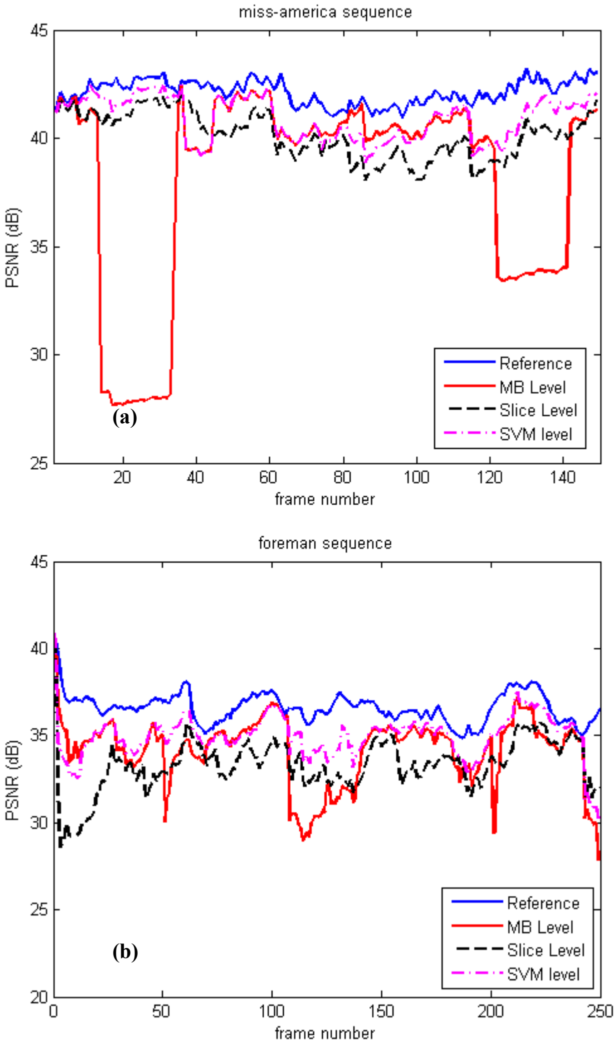


Fig. 4 (a) Miss-America (b) Foreman sequences at a BER of 1.00E-004

The superiority of our algorithm is even more evident subjectively. The MLC (Fig. 6a) 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 temporal domains. On the other hand, the SLC algorithm (Fig. 6b) is pessimistic, and although it manages to detect most of the errors at slice level, it conceals the whole slice even when only one MB is corrupted. This generally results in spatio-temporal propagation of unnecessarily concealed regions (e.g. the mouth of Miss-America in Fig. 6b is different from that in Fig. 6a because that MB was concealed even though it was not corrupted) which significantly reduce the perceptual quality of the reconstructed video sequence. The proposed algorithm detects only those MBs which provide major visual distortion at pixel level and thus ensures that only the severely corrupted MBs are concealed.

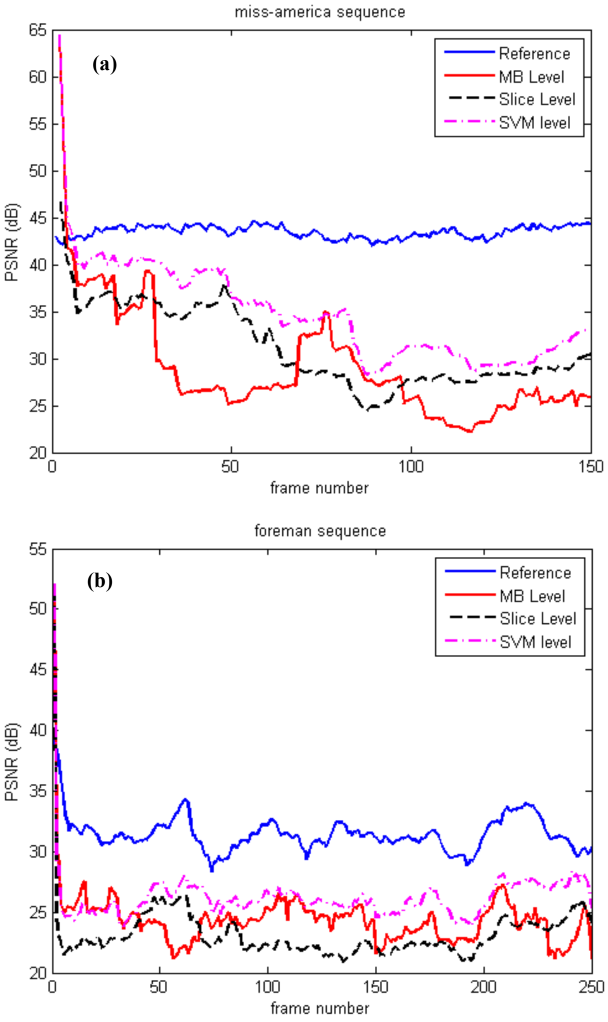


Fig. 5 (a) Miss-America (b) Foreman sequences at a BER of 5.00E-004

V. Comments and Conclusion

This paper has proposed a novel solution which enhances the error detection capabilities of the standard H.264/AVC video decoder. This new technique provides a good compromise between the MLC and SLC methods by actually locating those MBs which provide major artefacts in order to conceal them. In this way, uncorrupted MBs are not concealed, eliminating any unnecessary concealment and thus improving the quality of the reconstructed video.

The proposed solution significantly outperforms the SLC method generally adopted by the standard H.264/AVC decoder where PSNR gains of up to 5.74 dB were observed. Furthermore, the proposed solution manages to detect most of the visually distorted MBs and the ones that are not detected generally provide minimal visual distortions. This occurs because the SVMs were optimized to detect the major artefacts. The SVMs included by the system introduce minimal complexity to the system since the optimal hyperplane is represented by a small set of support vectors. Moreover, the proposed algorithm is only applied when there are errors on the channel and its computational complexity is low (less than 10% at high error rates) making it practical for applications such as video telephony and multicast/broadcast systems.

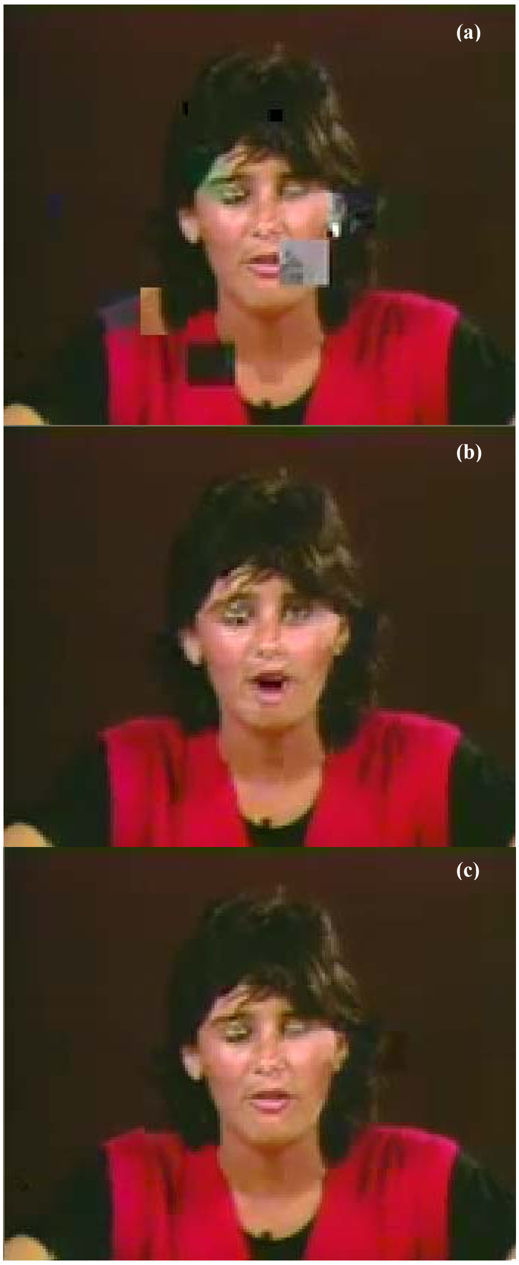


Fig. 6 Miss-America using (a) MB level concealment, (b) Slice Level Concealment, and (c) Proposed Error Detection method

References

[1] T. Stockhammer, M.M. 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] L. Superiori, O. Nemethova, and M. Rupp, "Performance of a H.264/AVC Error Detection Algorithm Based on syntax Analysis," in *Proc. of 4th Int. Conf. on Advances in Mobile Computing and Multimedia*, Yogyakarta, Indonesia, 2006, pp. 1-10.

[4] O. Nemethova, J.C. Rodriguez and M. Rupp, "Improved Detection for H.264 Encoded Video Sequences over Mobile Networks," in *IEEE Proc. of 8th Int. Symp. on Communication Theory and Applications*, Ambleside, Lake District, UK, 2005.

[5] O. Nemethova, G.C. Forte, and M. Rupp, "Robust Error Detection for H.264/AVC using Relation Based Fragile Watermarking," in *IEEE Proc. of 13th Conf. on Systems and Image Processing*, Budapest, Hungary, 2006.

[6] 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 on Multimedia Signal Processing*, Siena, Italy, Oct. 2004.

[7] 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 on Multimedia Signal Processing*, Siena, Italy, Oct. 2004.

[8] L. Superiori, O. Nemethova, and M. Rupp, "Detection of Visual Impairments in the Pixel Domain of Corrupted H.264/AVC Packets," in *IEEE Proc. of Int. Picture Coding Symposium*, Lisbon, Nov. 2007.

[9] R.A. Farrugia and C.J. Debono, "Enhancing Error Resilience in Wireless Transmitted Compressed Video Sequences through a Probabilistic Neural Network Core", in *IEEE Proc. of Int. Picture Coding Symposium*, Lisbon, Nov. 2007.

[10] R.A. Farrugia and C.J. Debono, "Resilient Transmission of H.264/AVC Video Sequences Using Probabilistic Neural Networks", in *Proc. 2008 Int. Symposium on Communications, Control and Signal Processing*, Malta, Mar. 2008.

[11] T. Ojala, and M. Pietikäinen, "Unsupervised texture segmentation using feature distributions," *Pattern Recognition 32*, pp. 477-486, 1999.

[12] N. Cristianini, and J.D. Taylor, *Support Vector Machines and other kernel-based learning methods*, Cambridge, UK:CUP, 2000.

[13] S.S. Keerthi, 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. 637-349, Mar. 2001.

[14] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in Large Margin Classifiers*, 1999.

[15] H.264/AVC Software Coordination, "JM Software." Ver. 12.2 [Online]. Available: http://iphone.hhi.de/suehring/html.