4

Resilient Digital Video Transmission over Wireless Channels using Pixel-Level Artefact Detection Mechanisms

Reuben A. Farrugia and Carl James Debono
University of Malta
Malta

1. Introduction

Recent advances in communications and video coding technology have brought multimedia communications into everyday life, where a variety of services and applications are being integrated within different devices such that multimedia content is provided everywhere and on any device. H.264/AVC provides a major advance on preceding video coding standards obtaining as much as twice the coding efficiency over these standards (Richardson I.E.G., 2003, Wiegand T. & Sullivan G.J., 2007). Furthermore, this new codec inserts video related information within network abstraction layer units (NALUs), which facilitates the transmission of H.264/AVC coded sequences over a variety of network environments (Stockhammer, T. & Hannuksela M.M., 2005) making it applicable for a broad range of applications such as TV broadcasting, mobile TV, video-on-demand, digital media storage, high definition TV, multimedia streaming and conversational applications.

Real-time wireless conversational and broadcast applications are particularly challenging as, in general, reliable delivery cannot be guaranteed (Stockhammer, T. & Hannuksela M.M., 2005). The H.264/AVC standard specifies several error resilient strategies to minimise the effect of transmission errors on the perceptual quality of the reconstructed video sequences. However, these methods assume a packet-loss scenario where the receiver discards and conceals all the video information contained within a corrupted NALU packet. This implies that the error resilient methods adopted by the standard operate at a lower bound since not all the information contained within a corrupted NALU packet is un-utilizable (Stockhammer, T. et al., 2003).

Decoding partially damaged bitstreams, where only corrupted MBs are concealed, may be advantageous over the standard approach. However, visually distorted regions which are not accurately detected by the syntax analysis of the decoder generally cause severe reduction in quality experienced by the end-user. This chapter investigates the application of pixel-level artefact detection mechanisms which can be employed to detect the visually impaired regions to be concealed. It further shows that heuristic thresholds are not applicable for these scenarios. On the other hand, applying machine learning methods such as Support Vector Machines (SVMs) can significantly increase the decoder's capability of detecting visual distorted regions. Simulation results will show that the SVMs manage to detect 94.6% of the visually impaired MBs resulting in Peak Signal-to-Noise (PSNR) gains of up to 10.59 dB on a

frame-by-frame basis. This method can be adopted in conjunction with other standard error resilient tools without affecting the transmission bit-rate required. Furthermore, the additional complexity is manageable which makes it applicable in real-time applications.

2. The effect of transmission errors

The H.264/AVC can achieve high compression efficiency with minimal loss of visual quality (Gonzalez, R.C., & Woods R.E., 2008). However, the resulting bitstream is susceptible to transmission errors, a phenomenon common in wireless environments, where even a single corrupted bit may cause disastrous quality degradation for an extensive period of time. This is mainly because video and image compression standards employ variable length codes (VLCs) to maximise the compression efficiency. Transmission errors may cause a decoder to lose synchronization and fail to decode subsequent VLC symbols correctly, as shown in Fig. 1. Thus, a single corrupted bit may result in a burst of corrupted pixels within the decoded frame. Furthermore, the spatial and temporal prediction algorithms adopted in block-based video compression standards such as H.264/AVC employ neighbouring macroblocks (MBs) and regions from reference frames respectively for prediction. If these regions are distorted, the reconstructed frame will be distorted as well, thus causing spatio-temporal propagation of errors (Richardson I.E.G., 2003).

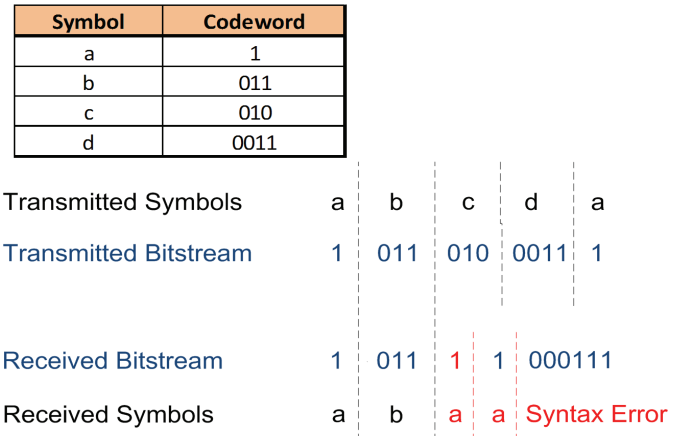

Fig. 1. Effect of transmission errors on variable length encoded sequences

In the standardization of H.264/AVC, corrupted packets are considered as being discarded by the receiver. Therefore, even a single corrupted bit within a packet will cause all the MBs contained within that packet to be dropped and concealed, thus, in general, the decoder will conceal a number of uncorrupted MBs. Furthermore, the spatio-temporal propagation of the superfluously concealed regions will propagate in both space and time, which by and large results in a significant reduction in perceptual quality. Fig. 2 illustrates the effect of a single corrupted bit in frame 41 of the *Foreman* sequence on the perceptual quality of the reconstructed frame and the effect of spatio-temporal propagation in the following frames.
Decoding of partially corrupted payloads may be beneficial for some applications. This is particularly true when considering damaged video bitstreams, since most of the MBs contained within a corrupted packet are either not corrupted or else provide imperceptible

Fig. 2. Error propagation in standard H.264/AVC caused by a single bit error

artefacts. Driven by this observation a set of protocols which allow the delivery of damaged packets are available. An overview of these methods is provided in (Welzl, M., 2005). However, as shown in Fig. 3, transition errors which are not detected by the syntax analysis of the H.264/AVC decoder may cause significant visual distortions which propagate in the spatio-temporal domain, thus reducing the end-user experience. The syntax analysis only manages to detect 57% of the corrupted MBs (Superiori, L. et al., 2006). Thus, to make decoding of partially damaged data applicable for video applications, the design of decoder-based algorithms which better cope with transmission errors without affecting the transmission bit rate are required.

3. Standard error resilient tools

The H.264/AVC video coding standard specifies several error resilient mechanisms aimed at minimizing the effect of transmission errors on the perceptual quality of the received video content. The following error resilient mechanisms are included in the standard:

- Slice Structuring
- Intra Refresh
- Flexible Macroblock Ordering (FMO)
- Redundant Slices (RS)
- Data Partitioning (DP)

3.1 Slice structuring

Delivery of video content over wireless networks is generally provided through frames having small maximum transmission units (MTU). This is mainly because the probability of

Fig. 3. Error propagation of artefacts caused by transmission errors when decoding partially damaged slices

bit errors affecting a small packet is lower than that for larger packets (Stockhammer, T. et al., 2003). Wireless video transmission is not an exception, where each frame is segmented into a number of independent coding units called slices. Slices are coded using limited spatial prediction and thus can be considered to provide spatially distinct synchronization points (Kumar, G. et al., 2006).

Furthermore, small packets reduce the amount of lost information, and hence the error concealment method can be applied to smaller regions, providing a better quality of experience. In (Kumar, G. et al., 2006) it was reported that slice structuring provides Peak Signal-to-Noise Ratio (PSNR) gains of around 7.33 dB when compared to single picture per packet with no error resilience.

3.2 Intra refresh

The application of slice structures limits the spatial propagation of errors within slice boundaries. However, due to the hybrid design of the H.264/AVC coding engine which adopts information from previously decoded frames, temporal propagation of artefacts caused by transmission errors is still severe. Instantaneous decoding refresh (IDR) pictures can be used to eliminate the temporal propagation of distorted regions where the entire picture is intra encoded. However, for real-time conversational video applications, it is not advisable to insert I-frames due to bit rate constraints and the resulting long delays involved (Kumar, G. et al., 2006).

Random intra MB coding is more acceptable for real-time applications. In addition to reducing the temporal propagation of distorted regions, it allows the encoder to maintain a

constant bit rate with the help of an H.264/AVC integrated rate-distortion control mechanism. Additionally, this technique is an encoder-based tool, which increases the data rate requirement but provides no additional overheads at the decoder.

3.3 Flexible Macroblock Ordering (FMO)

A more advanced error resilient tool specified in the H.264/AVC standard is the flexible macroblock ordering (FMO) which allows the specification of MB allocation within slice groups. The objective behind FMO is to scatter possible errors to the whole frame as equally as possible to avoid error clustering in a limited region. FMO is particularly powerful in conjunction with appropriate error concealment when the samples of a missing or corrupted slice are surrounded by many correctly decoded ones (Stockhammer, T. et al., 2003).
FMO can provide a significant gain in quality even at high error rates. However, FMO disallows the intra-frame prediction to exploit spatial redundancy in neighbouring MBs, since they are not enclosed within the same slice group. This will reduce the coding efficiency of the codec, thus limiting the applicability of FMO to low bit rate applications. Experimental results have shown that when transmitting over an error free channel, FMO encoded video sequences incur a data rate increase of 10% compared to when FMO is switched off (Wenger, S. & Horowitz, M., 2002).

3.4 Redundant Slices (RS)

A redundant slice is a new error resilient feature included within the H.264/AVC standard which allows the encoder to send redundant representations of various regions of pictures. Redundant slices may use different coding parameters, such as quantization levels, reference pictures, mode decisions, and motion vectors, than those used to encode the primary picture. If the primary slice is received correctly the redundant slice is simply discarded. However, if it is corrupted, the redundant slice is used instead in order to limit the visual distortion caused by transmission errors (Zhu, C. et al, 2006).
The enhanced error resilience provided by this strategy is achieved at the cost of additional overheads in terms of the excess bit rate requirements. There exists a trade-off between the degradation in the picture quality of the recovered video due to redundant slices and the available bandwidth. The more information introduced to describe the secondary slices the better is the performance of the decoder. Even though there is no restriction on the amount of information to be included in the redundant slices, in most applications bandwidth draws the limit and thus affects the performance of the redundant slices mechanisms.

3.5 Data Partitioning (DP)

The H.264/AVC codec generally encodes each frame to provide one single bitstream which forms a slice. However, since some coded information is more important than others, H.264/AVC enables the syntax of each slice to be separated into three different partitions for transmission (Wenger, S., 2003):

- Header information, including MB type, quantization parameters and motion vectors. This information is the most important, because without it, symbols of the other partitions cannot be used. This partition is called type A.
- The Intra partition (Type B) carries intra coded pictures and intra coefficients. This partition requires the availability of the type A partition of a given slice in order to be useful.

- The Inter partition contains only inter coded pictures and inter coefficients. This partition contains the least important information and in order to be useful requires the availability of the type A partition, but not the type B partition.

All partitions have to be available to execute standard conformant reconstruction of the video content. However, if the inter or intra partitions are missing, the available header information can still be used to improve the performance of the error concealment. Data partitioning is not included in the H.264's baseline profile and thus cannot be adopted for typical videoconferencing and mobile applications (Liu, L. et al., 2005).

4. Related work

Several extensions to the standard and novel error resilient strategies have been proposed in literature. Fragile watermarking was adopted in (Nemethova, O. et al., 2006), (Chen, M. et al., 2005) and (Park, W. & Jeon, B., 2002) to embed information that aids the detection and concealment of distorted regions. A low resolution version of each video frame was embedded in itself in (Adsumilli, C.B. et al., 2005) using spread-spectrum watermarking and is used to aid concealment of distorted regions. However, embedding information contributes to a reduction in the quality of the transmitted video content even when transmission is performed over an error free channel.

In order to better protect the transmitted bitstreams, error control strategies can be employed. Instead of including structured redundancy at the encoder, which reduces the compression efficiency, the authors in (Buttigieg, V. & Deguara, R., 2005) have replaced the VLC tables of the MPEG-4 video compression standard with variable length error correcting (VLEC) codes. A soft-decision sequential decoding algorithm was then implemented to decode the MPEG-4 bitstreams. However, the adoption of VLEC codes reduces the compression efficiency of the codec since the codewords produced have a longer average length.

The redundancy in compressed image and video was analyzed in (Nguyen, H. & Duhamel, P., 2003), where it was concluded that significant gain in performance can be achieved when additional video data properties are taken into consideration while decoding. The same authors have later adopted a modified Viterbi decoder algorithm to recover feasible H.263 video sequences in (Nguyen, H. & Duhamel, P., 2003), (Nguyen, H. & Duhamel. P. et. al., 2004). Sequential decoding methods were adopted for H.264/AVC encoded sequences where Context Adaptive Variable Length Codes (CAVLC) (Weidmann, C. et. al, 2004) and (Bergeron, C. & Lamy-Bergot, C., 2004) and Context Adaptive Binary Arithmetic Codes (Sabeva, G. et. al, 2006) coding modes were considered. However, sequential decoding algorithms introduce variable decoding delays which can be problematic in real-time applications (Lin, S. & Costello, D.J., 1983). A survey on joint-source channel coding techniques is provided in (Guillemot, C. & Siohan, P., 2005).

Various pixel-level artefact detection mechanisms based on heuristic thresholds were proposed in (Superiori, L. et. al, 2007), (Farrugia, R.A. & Debono, C.J., 2007) and (Ye, S., et. al, 2003). However, these methods only manage to detect between 40~60% of the corrupted MBs and have limited applications in practice since the optimal thresholds vary from sequence to sequence. An iterative solution which was presented in (Khan, et. al, 2004) attained substantial gain in quality at the expense of significantly increasing the complexity of the decoder, making it unsuitable for real-time mobile applications.

Scalable video coding (SVC) is another solution that can provide resilient delivery of video content over wireless channels (Schwarz, H. et. al., 2006). A scalable representation of video

consists of a base layer providing basic quality and multiple enhancement layers serving as a refinement of the base layer. The enhancement layers however are useless without the base layer. The benefit of SVC for wireless multiuser was demonstrated in (Liebl, G. et. al, 2006). However, this is achieved at the cost of higher encoding complexity and higher bit rate demand (Roodaki, H. et. al, 2008), (Ghanbari, M., 2003)).
Multiple Description Coding (MDC) (Goyal, V.K., 2001) is another alternative approach whose objective is to encode a source into two bitstreams such that high-quality reconstruction is derived when both bitstreams are received uncorrupted, while a lower but still acceptable quality reconstruction is achieved if only one stream is received uncorrupted. Several methods which adopt MDC to enhance the robustness of the video codec can be found in literature (Wang,Y. & Lin, S., 2002), (Tilio, T. et. al, 2008) and (Wang, Y. et. al, 2005). Again, the increase in error resilience is achieved by increasing the data rate required to deliver the same quality criterion as conventional single description coding methods in the absence of transmission errors.
Unequal Error Protection (UEP) schemes were extensively investigated in order to give higher protection to more important information. UEP was successfully combined with Data Partitioning (Barmada, B. et. al, 2005) and SVC (Zhang, C. & Xu, Y., 1999), (Wang, G. et. al 2001). Unbalanced MDC methods which allocate less bit rates to channels operating in bad conditions were proposed in (Ekmekci Flierl, S. et. al, 2005). The authors in (Rane, S. et. al, 2006) have proposed the transmission of a Wyner-Ziv encoded version of the image using the redundant slice option of H.264/AVC. Finally, Encoder-Decoder interactive error control approaches were presented in (Wang, J.T. & Chang, P.C., 1999), (Girod, B. & Färber, N., 1999) and (Budagavi, M. & Gibson, J.D., 2001). These methods however introduce additional delays making them unsuitable in wireless real-time applications and are not applicable for typical broadcasting/multicasting applications.

5. Pixel-level artefact detection mechanism

The standard H.264/AVC video coding standard was built with the assumption that corrupted slices are discarded and therefore does not allow partial decoding of corrupted payloads. This forces these mechanisms to operate at a lower bound since they assume a worst case scenario, where all the MBs contained within a corrupted slice are discarded and concealed. This assumption is most of the time untrue since in the majority of cases the MBs contained within a corrupted slice are either not corrupted or else provide imperceptible visual distortions. Therefore, decoding of partially damaged payloads may be beneficial when considering damaged compressed video content. For this reason, a set of syntax and semantic violations, presented in (Superiori, L. et al., 2006) were integrated within the H.264/AVC decoder to enable the decoding of damaged video content. However, these set of rules do not manage to accurately detect and localize a number of corrupted MBs resulting in severe visual impairments which propagate in the spatio-temporal domain, significantly reducing the quality of the reconstructed video sequences.
The pixel-level artefact detection mechanism is included as a post-process of the H.264/AVC sequences, as shown in Fig. 4, to detect the residual visually distorted regions to be concealed. The decoder is informed through the NALU header of the presence of transmission errors within the slice being decoded. To minimise computational complexity, the pixel-level artefact detection mechanism is only invoked to detect artefacts within corrupted slices. Therefore, no extra computation is required to decode uncorrupted slices.

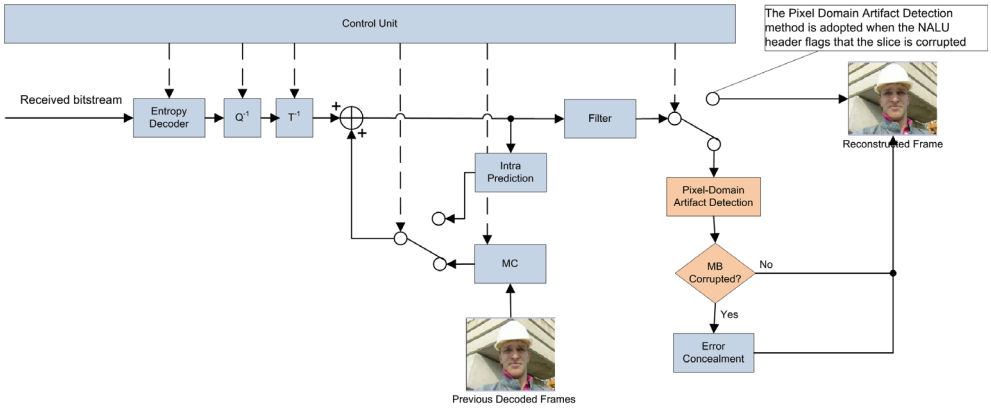

Fig. 4. Modified H.264/AVC Decoding Strategy using Pixel-Level Artefact Detection

The pixel-level artefact detection mechanism exploits the spatio-temporal smoothness of video scenes to detect visually impaired MBs. A number of dissimilarity metrics which exploit the redundancy present at pixel level are considered in subsection 5.1. These dissimilarity metrics generally provide large values to corrupted MBs and small values to uncorrupted MBs. Heuristic thresholds can be applied to discriminate between corrupted and uncorrupted MBs. However, the selection of these thresholds is sequence dependent and is thus not flexible enough to cater for a wide range of video sequences. This section introduces the application where a set of dissimilarity metrics were combined to form a feature vector, which is used to describe the reliability of each MB contained within a corrupted slice. The discrimination between corrupted and uncorrupted MBs is then provided through supervised machine learning algorithms which are discussed in subsection 5.2. These classifiers are trained and evaluated using subjective results provided through a case study discussed in subsection 5.3.

5.1 Dissimilarity metrics

5.1.1 Average Inter-sample Difference across Boundaries (*AIDB*)

In an image, there exists sufficient similarity among adjacent pixels, and hence across MB boundaries, even in the presence of edges. The *AIDB* dissimilarity metric is used to detect artefacts which affect the entire MB. Considering Fig. 5, let *M* denote a potentially corrupted MB under test with its four neighbouring MBs *N*, *S*, *E* and *W*. Further, let $p^{in} = \{p_1^{in}, p_2^{in}, ..., p_K^{in}\}$ represent boundary pixels inside the MB *M* and $p^{out} = \{p_1^{out}, p_2^{out}, ..., p_K^{out}\}$ represent boundary pixels in one of the neighbouring MBs $X \in \{N, S, E, W\}$. Then, the *AIDB*(*M*:*X*) distance measure is given by:

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

where *K* is the size of the MB and $\|\bullet\|_2$ is the Euclidean distance. The *AIDB* dissimilarity metric is then computed by evaluating the average of *AIDB*(*M*:*X*) over the available neighbouring MBs.

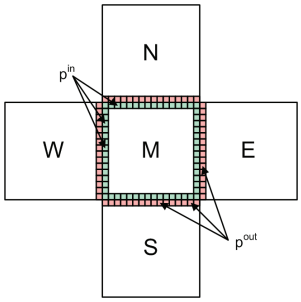

Fig. 5. Graphical representation of the $AIDB/IAIDB_{block}$ Dissimilarity Metrics

5.1.2 Internal AIDB per block ($IAIDB_{block}$)

The $IAIDB_{block}$ dissimilarity metric is based on the principle that, in non-corrupted MBs, the pixel transition from one 4×4 block to the adjacent block is generally smooth and thus the pixels at 4×4 boundaries are very similar. Opposed to the *AIDB* dissimilarity metric, the $IAIDB_{block}$ was designed to detect artefacts within an MB. For this purpose, each MB is divided into a grid of 16 4×4 blocks and the $IAIDB_{block}(M{:}X)$ metric is computed using (1), but this time the parameter M represents the 4×4 block within an MB under test and K = 4. The $IAIDB_{block}$ dissimilarity metric of each 4×4 block is then derived by averaging the $IAIDB_{block}(M{:}X)$ over the available neighbouring 4×4 blocks. At the end of the computation we have a set of 16 $IAIDB_{block}$ dissimilarity metrics.

5.1.3 Internal AIDB (*IAIDB*)

The *IAIDB* dissimilarity metric is based on the same spatial smoothness property described above. However, during experimentation, it was observed that some of the artefacts provide internal vertical/horizontal boundary discontinuities which could not be classified by the spatial features described above. The *IAIDB* dissimilarity metric is designed to provide a measure of the dissimilarity across the internal horizontal/vertical boundaries, as illustrated in Fig. 6. The metric is computed using:

$$IAIDB_{h/v} = \frac{1}{K}\left\| p^{in}_{h/v} - p^{out}_{h/v} \right\|_2 \tag{2}$$

where K = 16, $p_{h/v}$ represent the horizontal/vertical boundary pixels and $\|\bullet\|_2$ is the Euclidean distance.

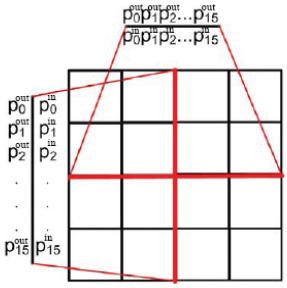

Fig. 6. Graphical representation of the *IAIDB* Dissimilarity Metric

5.1.4 Average Internal Difference between Subsequent Blocks (*AIDSB*)

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. The *AIDSB* dissimilarity metric for each 4×4 block b_t and b_{t-1}, shown in Fig. 7, is computed using:

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

where K represents the size of the block (in this case K = 4), $\left\|\bullet\right\|_2$ is the Euclidean distance, and p_t and p_{t-1} represent the pixel of the 4×4 block under test b_t and the corresponding block in the previous MB b_{t-1} respectively. Once the computation is terminated, a set of 16 *AIDSB* dissimilarity metrics, one for each 4×4, is available.

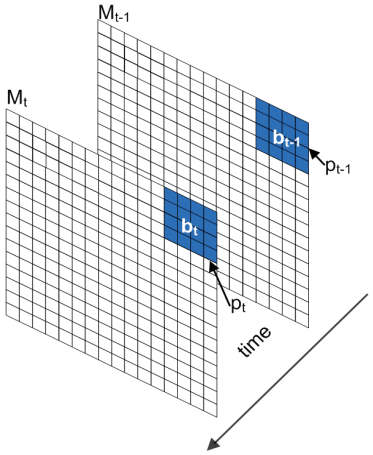

Fig. 7. Graphical representation of the *AIDSB* Dissimilarity Metric

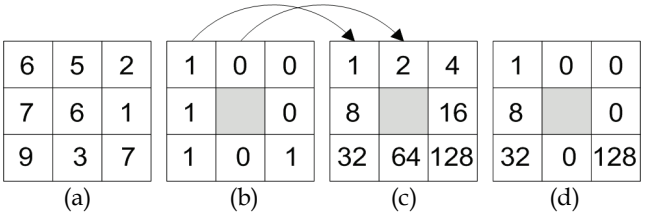

Fig. 8. Computation of the Local Binary Pattern

5.1.5 Texture Consistency (TC)

The Local Binary Pattern (LBP) operator (Ojala, T., et. al, 1996) is a powerful grey-scale invariant texture measure. To understand how it works let us consider Fig. 8. The original 3x3 neighbourhood (Fig. 8(a)) is thresholded by the value of the centre pixel. The values of the pixels in the thresholded neighbourhood (Fig. 8(b)) are multiplied by the binomial

weights given to the corresponding pixels (Fig. 8(c)). The result of this example is illustrated in Fig. 8(d). Finally, the values of the eight pixels are summed to obtain the LBP metric.
The LBP histograms of the current MB, h_t, and the corresponding MB in the previous frame h_{t-1} are first computed. The *TC* dissimilarity metric is then computed by evaluating the histogram insertion method given by:

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

where B is the number of bins which is set to 256.

5.1.6 Feature vector

The dissimilarity metrics described above exploit both colour and texture consistencies of the MB under test. This set can be reduced to eight dissimilarity metrics without compromising the performance of the classifiers described in the following section. These are:

1. *AIDB*
2. Mean of $IAIDB_{block}$
3. Standard Deviation of $IAIDB_{block}$
4. Vertical *IAIDB*
5. Horizontal *IAIDB*
6. Mean of *AIDSB*
7. Standard Deviation of *AIDSB*
8. *TC*.

These dissimilarity metrics are then combined together to form the feature vector, which solely describes the reliability of the MB under test. After extensive simulation and testing it was found that these eight dissimilarity metrics provide the best compromise between complexity and performance and therefore adopting higher dimensional feature vector resulted futile.

5.2 Classification methods

Pattern recognition techniques are used to classify some objects into one of the pre-defined set of categories or classes *c*. For a specific pattern classification problem, a classifier is developed so that objects are classified correctly with reasonably good accuracy. Inputs to the classifier are called features, which are composed of vectors that describe the objects to be classified. The features are designed according to the problem to be solved.
The aim of the pixel-level artefact detection mechanism is to detect the visually distorted MBs to be concealed. The feature extraction module extracts the feature vector which solely defines the reliability of the MB under test. The pattern recognition method then tries to categorize the MB under test in one of the categories based on the statistical information made available by the feature vector.
One of the simplest classification methods is to derive the probability density functions (PDFs) of the dissimilarity metrics representing the uncorrupted and corrupted MBs. The aim of the dissimilarity metrics is to have PDFs similar to the one illustrated in Fig. 9, at which point heuristic thresholds can be employed to detect visually distorted MBs. However, as it will be shown in the simulation results, these dissimilarity metrics provide

limited discriminative power when applied on their own. Furthermore, the distribution of uncorrupted MBs varies with varying video sequences and thus the optimal threshold is sequence dependent.

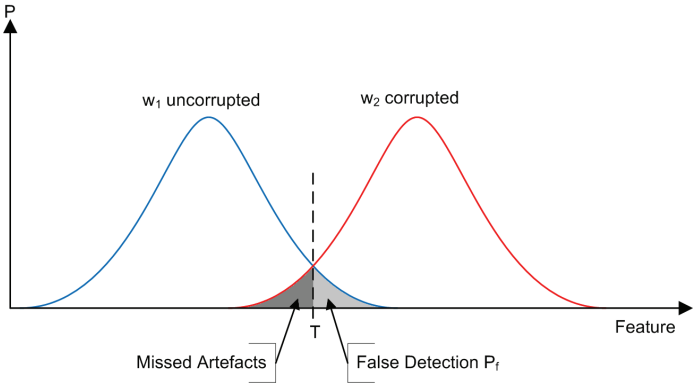

Fig. 9. Probability Density Function of a typical Dissimilarity Metric

Supervised learning classification methods can be employed to solve this problem. These algorithms adopt a set of l training samples $(\mathbf{x}, y)$, where $\mathbf{x} = [x_1, x_1, ..., x_n]$ represent the extracted feature vectors, y their corresponding labels (corrupted MB = -1, uncorrupted MB = +1), and n represents the number of dimensions of the feature vector. In supervised learning algorithms, the classifier attempts to learn the input/output functionality from examples to derive an optimal hyperplane to discriminate between the two classes.

5.2.1 Backpropagation Neural Network (*BPNN*)

A feed-forward neural network is a simple structure where multiple hidden layers are employed. Since it has the potential of approximating a general class of nonlinear functions with a desirable degree of accuracy, it has been employed in many pattern recognition applications (Gupta, M. M. et. al, 2003), (Duda R. O. et. al, 2000). The architecture of the feed-forward neural network is illustrated in Fig. 10.

Every neuron in the hidden layer receives an input vector $\mathbf{x}$. The output of all the neurons in the hidden layer, represented by a p-dimensional vector $\mathbf{z} = [z_1, z_1, ..., z_p]$ is fed forward to the neurons in the output layer. The output neurons generate an output vector $\mathbf{f} = [f_1, f_2, ..., f_m]$. Further, consider the weights corresponding to the i^{th} neuron in the hidden layer to be $\mathbf{w_i^{(1)}} = [w_{i1}^{(1)}, w_{i2}^{(1)}, ..., w_{in}^{(1)}], i = 1, 2, ..., p$ and the weights corresponding to the j^{th} neuron in the output layer to be $\mathbf{w_j^{(2)}} = [w_{j1}^{(2)}, w_{j2}^{(2)}, ..., w_{jp}^{(2)}]^T, j = 1, 2, ..., m$. The input/output relation of the neurons in the network can be expressed as:

$$\text{hidden layer} \begin{cases} s_i^{(1)} = \sum_{k=0}^{n} w_{ik}^{(1)} x_k \\ z_i = \sigma\left(s_i^{(1)}\right) \\ i = 1, 2, ..., p \end{cases} \quad \text{output neuron} \begin{cases} s_j^{(2)} = \sum_{q=0}^{p} w_{jq}^{(2)} z_q \\ f_j = \sigma\left(s_j^{(2)}\right) \\ j = 1, 2, ..., m \end{cases} \tag{5}$$

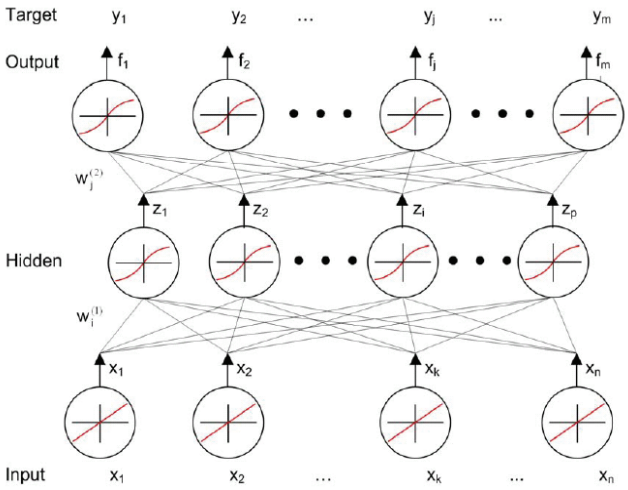

Fig. 10. Fully Connected Feed-Forward Neural Network

where $\sigma(\bullet)$ is a nonlinear activation function which is normally modelled by the sigmoid function as given by:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{6}$$

One of the most popular methods for training feed-forward neural networks is the backpropagation neural network (Rumelhart, D.E., et. al, 1986) (BPNN). The basic approach in learning starts with an untrained feed-forward neural network which is presented with an input training patter $\mathbf{x}$ and the corresponding output targets y. The BPNN algorithm then derives the weights which provide the best separating hyperplane which minimises the error function given by:

$$E = \frac{1}{2}\sum_{j=1}^{m}\left[y_j - f_j\right]^2 \tag{7}$$

where E is the cost function and is the measure of the learning performance of the network. The BPNN algorithm is an iterative method which adapts the weight vectors in the direction of decreasing error E (gradient decent) with respect to the weight vectors. The predicted weight difference is given by:

$$\Delta w_i^{(1)} = -\eta \frac{\partial E}{\partial w_i^{(1)}}, i = 1,2,...,p \tag{8}$$

$$\Delta w_j^{(2)} = -\eta \frac{\partial E}{\partial w_j^{(2)}}, j = 1,2,...,m \tag{9}$$

where $0 < \eta < 1$ is the learning rate constant. The weight adapting formulae for the hidden and output layer are as follows:

$$w_i^{(1)}(t+1) = w_i^{(1)}(t) + \eta\sigma'\left(s_i^{(1)}(t)\right)x(t)\sum_{l=1}^{m}\delta_l^{(2)}(t)w_l^{(2)}(t) \tag{10}$$

$$w_j^{(2)}(t+1) = w_j^{(2)}(t) + \eta\left[y_j(t) - f_j(t)\right]\sigma'\left(s_j^{(2)}(t)\right)z(t) \tag{11}$$

where

$$\delta_j^{(2)}(t) = \left(y_j(t) - f_j(t)\right)\sigma'\left(s_j^{(2)}(t)\right) \tag{12}$$

The design of the BPNN to solve the artefact detection problem is not trivial. Whereas the number of inputs and outputs is given by the feature space and number of categories respectively, the total number of hidden neurons in the network is not that easily defined. After extensive simulation and testing the best recognition rate was registered when applying a single hidden-layer of 50 neurons and applying a learning rate η = 0.1.

5.2.2 Probabilistic Neural Network (*PNN*)

The Probabilistic Neural Network (Specht, D.F., 1988) (PNN) is another feed-forward neural network commonly used for pattern recognition. The PNN classifier forms a Parzen estimate based on l samples, where each sample is represented by a normalized n-dimensional feature vector. The PNN architecture, as illustrated in Fig. 11, consists of n input units, where each unit is connected to each of the l pattern units. Each pattern unit is, in turn, connected to one and only one of the category units. The connections from the input to pattern units represent modifiable weights, which will be derived during the training phase. On the other hand, the connections between the pattern units and the output units have weights which are set to unity.

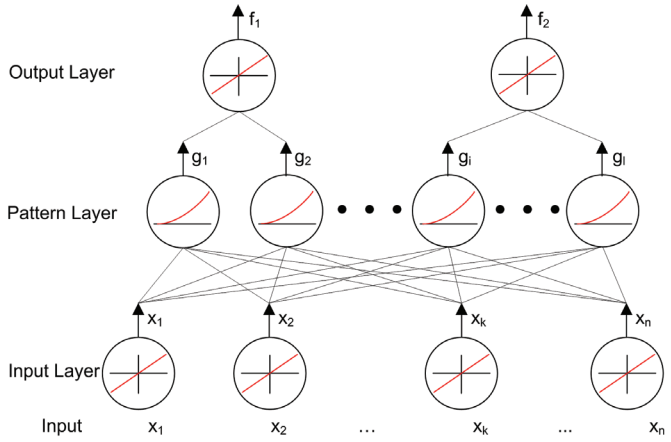

Fig. 11. Architecture of a Probabilistic Neural Network

The training of a PNN is quite straightforward, where at first each pattern **x** of the l training feature vectors is normalized to have unit length. The first normalized training pattern is then placed on the input units. The modifiable weights linking the input units and the first pattern unit are set such that $w_1 = x_1$. A single connection from the first pattern unit to the output unit corresponding to the known class of that training pattern is created. The process

is repeated for all the remaining training patterns, setting the weights to the successive pattern units such that $w_k = x_k$ for $k = 1, 2, ..., l$. After this training, we have a network that is fully connected between the input and pattern units, and sparsely connected from pattern to category units.
The trained PNN can now be used for classification by computing the inner product between the inputted feature vector **x** and the weight vector w of the k^{th} pattern unit as follows:

$$z_k = \langle w_k, x \rangle \tag{13}$$

where $\langle \bullet \rangle$ denotes the inner product operator. Each pattern unit emits a nonlinear activation function given by:

$$g_k = \exp\left(\frac{(z_k - 1)}{\sigma^2} \right) \tag{14}$$

where σ is the smoothing parameter, which after extensive simulation and testing was set to 0.12. Each output unit then accumulates the values of these activation functions of a given class. The feature vector is simply categorised with the largest output neuron value.

5.2.3 Support Vector Machine (*SVM*)

A Support Vector Machine (Cortes, C. & Vapnik, V., 1995) (SVM) is a very powerful method that in the few years since its inception has outperformed most other machine learning algorithms in a wide variety of applications. The aim of SVM classification is to derive a separating hyperplane which optimises the generalisation bounds. The generalisation theory gives clear guidance on how to control capacity and hence prevent overfitting by controlling the hyperplane margin measures, while optimisation theory provides the mathematical techniques necessary to find the hyperplane which optimises these measures.
The SVM classifier is a linear learning machine and therefore it may not manage to classify nonlinearly separable data at an acceptable accuracy. Thus, to enhance linear separability, the SVM employs an 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 optimised separating hyperplane. Several algorithms (Platt, J.C., 1998), (Keerthi, S.S., et. al, 2001) can be found in literature which can be used to train the SVM. These methods try to solve the following quadratic optimisation problem:

$$\begin{aligned} &\max_{\alpha} W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j K(\vec{x}_i, \vec{x}_j) \alpha_i \alpha_j, \\ &0 \le \alpha_i \le C, \forall i, \\ &\sum_{i=1}^{l} y_i \alpha_i = 0 \end{aligned} \tag{15}$$

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 decision function is given by:

$$f(x) = sign\left(\sum_{i \in SV} \alpha_i y_i K(\vec{x}_i, \vec{x}) + b\right) \tag{16}$$

where b is the bias term. Following this, the unknown data are classified using the decision function as follows:

$$x \in \begin{cases} \text{Class 1} & \text{if } f(x) > 0 \\ \text{Class 2} & \text{if } f(x) < 0 \\ \text{Unclassifiable} & \text{if } f(x) = 0 \end{cases} \tag{17}$$

For the application at hand, a modified version of the Sequential Minimal Optimization (SMO) algorithm was adopted (Keerthi, S.S., et. al, 2001) to train the SVM classifier. This algorithm was reported to be faster and provides better convergence when compared to the other methods (Platt, J.C., 1999). This classifier utilises the Radial Basis Function (RBF) Kernel which is given by:

$$K(x, y) = \exp\left(\frac{-\|x - y\|^2}{2\sigma^2}\right) \tag{18}$$

where σ is the smoothing parameter. Following extensive testing, the penalisation constant C was set to 60 while the smoothing parameter was set to 1.5.

5.3 Training the classification methods

The syntax analysis check rules used to detect syntax and semantic violation in the H.264/AVC bitstreams only manage to detect 57% of the transmission errors. The errors which do not cause syntax or semantic violations produce different levels of visually impaired regions. As shown in Fig. 12, some of the distorted MBs are very annoying while others are imperceptible. Most of the artefacts caused by transmission errors generally provide imperceptible visual distortions and thus concealing all the MBs contained within a corrupted slice results in concealing a number of MBs which provide minimal or no visual distortion.

Fig. 12. Typical residual artefacts undetected by the syntax analysis rules

In natural video sequences there exists sufficient correlation between spatio-temporal neighbouring MBs. As shown in Fig. 12, the statistics of annoying artefacts significantly

differ from those of uncorrupted MBs. This observation suggests that the design of an artefact detection mechanism which exploits the spatio-temporal redundancies available at pixel level after decoding can be used to detect distorted regions. The aim of the designed method is to maximise the detection of highly distorted MBs which significantly degrade the quality of the decoded frame while being more lenient with imperceptible ones.

To design a robust artefact detection mechanism a set of five video sequences (*Foreman*, *Carphone*, *Mobile*, *Coastguard* and *News*) were used to derive the training and testing data to be used for experimentation. A third set, called the cross-validation set, was made up from another four video sequences (*Miss-America*, *Salesman*, *Akiyo* and *Silent*). These sequences were encoded at QCIF resolution at 30 frames per second and were transmitted over a Binary Symmetric Channel (BSC) at different error rates. From the resulting distorted frames, a population of 3000 MBs is extracted at random. This population is divided into three distinct groups with each group consisting of 500 corrupted MBs and 500 uncorrupted MBs. The first group is used for training while the remaining two groups are used for recognition and cross-validation respectively.

To better analyze the performance of the artefact detection method, the distorted MBs were scaled according to the five-level distortion scale given in Table 1. The experiment consisted of 21 reliable viewers who did not have any experience in image quality evaluation. A subset of 74 corrupted MBs were chosen at random from the training set and were supplied to the viewers for assessment. The viewers have classified these MBs using a methodology similar to the single stimulus test (ITU-T Rec. P.910, 1999). Two experts in the area of multimedia communications have scaled the same images and were used as a reference.

Distortion Level (DL)	**Definition**
4	Very annoying artefacts
3	Annoying artefacts
2	Slightly annoying artefacts
1	Perceptible but non annoying artefacts
0	Imperceptible or non-corrupted MB

Table 1. Definition of the Distortion Levels

Applying the single stimulus test methodology to all the 1500 distorted MBs was prohibitive due to excessive time required. It is much less time consuming to derive this subjective evaluation through the judgement of a group of experts. However, before doing so, it had to be ensured that the results provided by the group of experts represented the opinion of normal users. For this reason, the One-Sample *t*-test was used to compare the opinions provided by the viewers to the reference results based on the subset of 74 corrupted images. This test confirmed that the difference between the means for 83.74% of the images considered during the test is not significant at a 95% confidence level. Furthermore, all the remaining samples have a negative mean difference indicating the judgement provided by the group of experts caters for the most demanding users.

6. Simulation results

The performance of the classification methods employed by the pixel-level artefact detection module is dependent on the dissimilarity metrics. The aim of the dissimilarity metrics employed is to provide small metrics for uncorrupted MBs and large metrics for corrupted

MBs in order to increase the separability in the input space, thus facilitating classification. Since these dissimilarity metrics (except the Texture Consistency) measure colour differences, it becomes clear that the performance of the dissimilarity metrics and thus of the classification method is dependent on the colour space model where these dissimilarity metrics are computed.

To identify the colour space model where to compute the dissimilarity metrics, the *AIDSB* and *AIDB* dissimilarity metrics are considered. These distance measures are computed in the *YUV*, *HSI*, *CIELAB*, and *CIELUV* colour systems, where the heuristic thresholds T are derived using the training set. The heuristic thresholds are selected in such a way that we achieve an acceptable error detection rate P_D at a false detection rate P_F of around 5%. The dissimilarity based classification method is then evaluated based on the data contained in the training set and the results obtained are summarised in Table 2 and Table 3.

Colour System	**Threshold T**	P_D **(%)**	P_F **(%)**
YUV	7.50	65.20	5.20
HSI	0.25	71.80	5.60
CIELAB	-0.10	68.00	6.60
CIELUV	0.25	72.80	4.60

Table 2. Performance of the *AIDSB* dissimilarity metric using different colour systems

Colour System	**Threshold T**	P_D **(%)**	P_F **(%)**
YUV	8.00	31.40	4.60
HSI	0.30	40.20	4.60
CIELAB	-0.20	34.00	5.40
CIELUV	0.20	41.80	4.20

Table 3. Performance of the *AIDB* dissimilarity metric using different colour systems

These results evidence that the dissimilarity-based classification methods perform better when computed in colour systems which better describe the human perception. In particular, computing these dissimilarity measures in the perceptually uniform CIELUV colour space model, generally adopted in television and video display applications, seems to be more beneficial. For this reason, in the remaining part of this section the dissimilarity metrics (except Texture Consistency) are computed in the CIELUV colour space model.

The three supervised learning algorithms described in the previous section can now be trained on the training set using the eight dissimilarity metrics described above to represent each component of the feature vector. The performance of these algorithms is compared to the *AIDB* and *AIDSB* based classifiers and the results obtained are summarized in Table 4. From these results, it can be concluded that the three supervised learning algorithms manage to detect all severely distorted MBs whereas dissimilarity-based classifiers do not. Furthermore, an overall recognition gain of around 20% and 50% is achieved relative to the *AIDSB* and the *AIDB* based classifiers, respectively. Finally, it can be observed that the SVM achieves the best result, where it manages to detect 94.6% of the visual artefacts at P_F smaller than 5%. The gain achieved by the SVM classification method over the other neural approaches occurs mainly because this technique manages to detect more DL2 and DL1 artefacts.

Classifier	P_D (%)	P_{DL4} (%)	P_{DL3} (%)	P_{DL2} (%)	P_{DL1} (%)	P_F (%)
AIDB	41.80	86.57	45.32	18.18	2.74	4.20
AIDSB	72.80	97.76	87.77	62.34	21.55	4.60
BPNN	92.60	100.00	100.00	93.51	63.01	7.40
PNN	92.20	100.00	100.00	92.21	63.01	4.80
SVM	94.60	100.00	100.00	94.81	73.97	4.60

Table 4. Performance of the different classification methods

Both SVM and PNN solutions achieve good artefact detection capabilities with the SVM classifier performing best. However, in order to avoid overfitting, these classifiers are tested on a cross-validation set which contains feature vectors extracted from different video sequences than the ones considered in the training phase. The results are summarised in Table 5, where it can be noticed that the SVM still performs well on these video sequences while the performance of the PNN degrades significantly. Another point in favour of the SVM classifier is that, as shown in Fig. 13, the classification is less computational intensive. This is attributed to the fact that only 156 support vectors are used to derive the separating hyperplane. Given these results, the SVM was integrated within the pixel-level artefact detection method.

Classifier	P_D (%)	P_{DL4} (%)	P_{DL3} (%)	P_{DL2} (%)	P_{DL1} (%)	P_F (%)
PNN	78.89	100.00	96.23	67.69	65.45	4.80
SVM	90.95	100.00	100.00	90.77	78.18	5.20

Table 5. Performance of the different classification methods on cross-validation set

The pixel-level artefact detection module is then integrated within the JM software model, where the syntax and semantic violation test procedures are enabled to allow the decoding of partially damaged H.264/AVC bitstreams. The raw video sequences are encoded at QCIF resolution at 15 frames per second at a data rate of 64 kbps with the format IPPP.... The encoder employs slice structuring, where each slice is forced to a size strictly smaller than 100 bytes. The resulting slices are encapsulated within RTP/UDP/IP packets and transmitted over an AWGN channel at different noise levels.

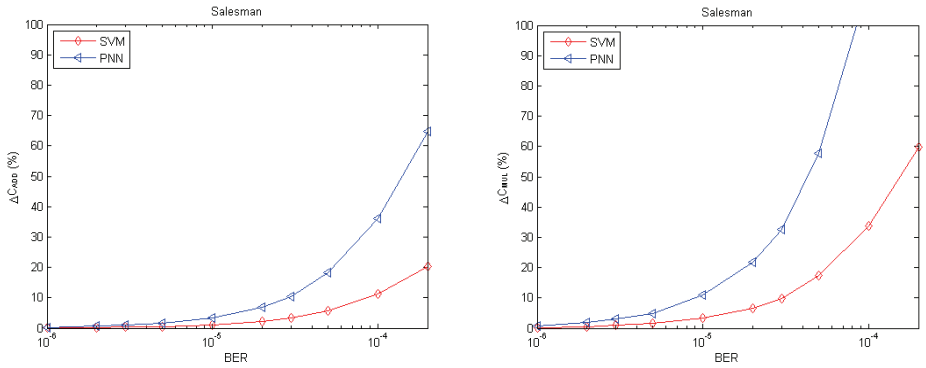

Fig. 13. Complexity analysis of the SVM and PNN methods (left) ΔC_{Add} and (right) ΔC_{Mul} for the *Salesman* sequence

The pixel-level artefact detection method is tested on the *Salesman* video sequence, which is not used during the training phase. The performance of the standard decoder and the modified decoder which employs an SVM classifier at its core is shown in Fig. 14. These results confirm that concealing only those MBs which provide visually distorted regions is beneficial over the standard decoding method, where the PSNR gains over the whole sequence are higher than 0.5 dB at moderate to high error rates. Additionally, PSNR gains of up to 3.90 dB are observed. This superior performance is attributed to the fact that the pixel-level artefact detection method localises the MBs which need to be concealed and thus the over-concealment problem is minimised. Furthermore, reducing the area to be concealed results in improved performance of the error concealment method employed.

The gain achieved by the pixel-level artefact detection method is consistent even when using video sequences which contain fast moving objects. However, in such sequences, it was observed that in the presence of abrupt shots a number of false detections occur which force undistorted MBs to be concealed. Since this method is only employed on distorted slices, the pixel-level artefact detection method will perform at worst like the standard decoder (i.e. detecting all the MBs contained within a slice to be corrupted) hence such cases do not deteriorate the performance compared to the standard.

The performance of the pixel-level artefact detection method is further tested on a frame-by-frame basis and the results are provided in Fig.15 – Fig. 16. These results confirm the superiority of the pixel-level artefact detection method which employs the SVM classifier at its core. The PSNR on a frame-by-frame basis of this method is consistently superior to the standard decoder, where PSNR gains of up to 10.59 dB are achieved. The gain in subjective quality is even more impressive, where it can be seen that concealing undistorted regions provides artefacts which reduce the quality of the video sequence. On the other hand, the pixel-level artefact detection method manages to reconstruct the video sequence at an acceptable level of quality which is quite similar to the original undistorted video sequence.

The artefact detection method can also be employed in conjunction with other standard error-resilient tools such as intra-refresh and FMO. As shown in Fig. 17 and Fig. 18 the pixel-level artefact detection method boosts the performance of the standard error resilient tools by more than 0.3 dB in PSNR at moderate to high error rates, where PSNR gains of up to 2.08 dB are observed. The flexibility of the proposed solution is attributed to the designed features which are used by the classifier to detect distorted regions and the generalisation achieved by the SVM classifier.

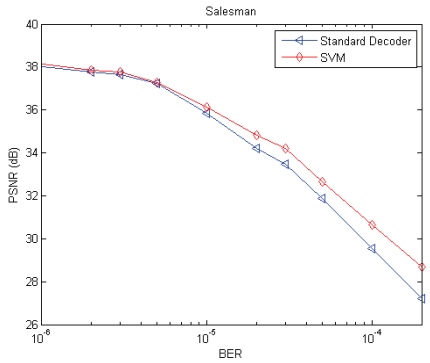

Fig. 14. Performance of the Pixel-Level Artefact-Detection method for the *Salesman* sequence

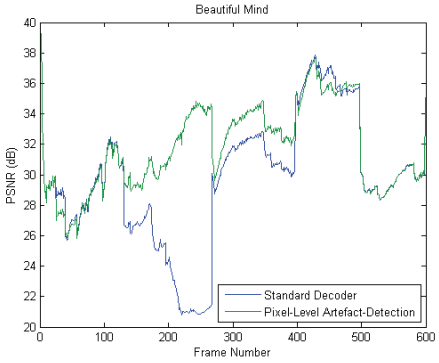

Fig. 15. Performance of this method at a BER of 1.00E-005 for the *Beautiful Mind* sequence

(a) (b) (c)

Fig. 16. Frame 215 from the sequence *Beautiful Mind* at 64 kbps (a) reference sequence without errors, (b) standard decoder, and (c) Pixel-Level Artefact Detection method

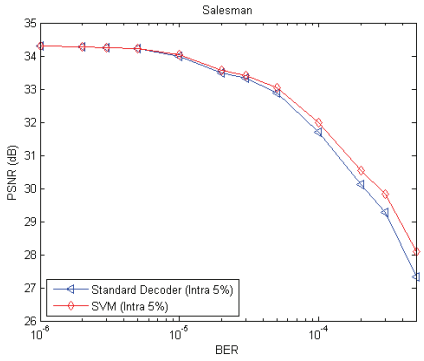

Fig. 17. Performance of the Pixel-Level Artefact-Detection method using Intra Refresh 5% for the *Salesman* sequence

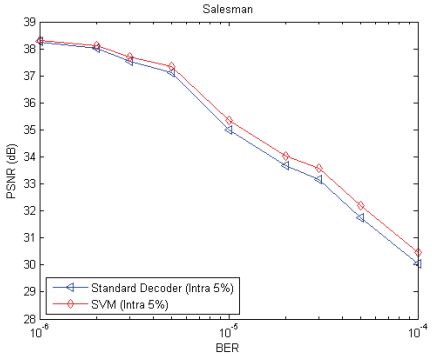

Fig. 18. Performance of the Pixel-Level Artefact-Detection method using Dispersed FMO for the *Salesman* sequence

7. Comments and conclusion

Decoding of partially damaged H.264/AVC generally results in distorted regions which severely affect the quality of the reconstructed video sequence. In this chapter, it was shown that robust pixel-level artefact detection methods can be used to detect those distorted MBs which provide major visual distortion, which are later concealed. In this way, only visually impaired regions are concealed by the decoder resulting in an improved quality of experience. Several classification methods have been considered to solve this problem. After several testing and cross-validation, it is concluded that the SVM classifier achieves the best performance, where it manages to detect 94.6% of the visually distorted regions at a false detection rate of around 5%. More importantly, it provides an unequal important artefact detection strategy, where all annoying artefacts (DL4, DL3) are detected but at the same time the solution is more lenient with slightly annoying artefacts (DL2, DL1).

The method discussed in this chapter makes the H.264/AVC decoder more resilient to transmission errors with overall PSNR gains higher than 0.5 dB being observed at high error rates. This increased robustness does not incur additional bit-rate. In fact the algorithm is computed entirely at the decoder, where it exploits the inherent redundancies available at pixel-level between spatio-temporal neighbouring MBs to detect distorted MBs. Furthermore, the complexity introduced by this method is manageable even at high error-rates making it more applicable to real-time mobile applications. This method can be further applied in conjunction with other error-resilient methods adopted by the standard to boost their performance. Moreover, it is possible to extend this concept to decode other block-based video coding systems such as H.263.

The performance of the classifier adopted by the pixel-level artefact detection method is dependent on the generalisation achieved by the classifier. During testing it was confirmed that the SVM classifier did generalise better than the other classification methods especially when considering video sequences which were not used during the training phase. In fact, the performance of the SVM classifier has managed to outperform the PNN classifier in the cross-validation tests for the H.264/AVC encoded sequences.

During the experiments it was noticed that the method suffered in presence of abrupt screen changes. In fact, due to the temporal dissimilarity metrics, the SVM classifies most of the

MBs affected by the scene change to be artefacts and thus are concealed. However, since the SVM classifier is adopted only on damaged slices, at worst the pixel-level artefact detection method will perform like the standard H.264/AVC implementation where all MBs contained within a corrupted slice are concealed.

8. References

Adsumilli, C.B., Farias, M.C.Q., Mitra, S.K. & Carli, M., 2005. A Robust Error Concealment Technique using Data Hiding for Image and Video Transmission over Lossy Channels, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 15, no. 11, Nov. 2005, pp. 1394-1406, ISBN 1051-8215

Barmada, B., Ghandi, M.M., Jones, E.V. & Ghanbari, M. (2005). Prioritized Transmission of Data Partitioned H.264 Video with Hierarchical QAM, *IEEE Signal Processing Letters*, Vol. 12, no. 8, Aug. 2005, pp. 577-580, ISSN 1070-9908

Bergeron, C. & Lamy-Bergot, C., (2004), Soft-Input Decoding of Variable-Length Codes applied t the H.264 Standard, *IEEE Workshop on Multimedia Signal Processing*, Siena, Italy, 2004

Budagavi, M. & Gibson, J.D., (2001). Multiframe Video Coding for Improved Performance over Wireless Channels, *IEEE Transactions on Image Processing*, Vol. 10, no. 2, Feb. 2001, pp. 252-265, ISBN 1057-7149

Buttigieg, V. & Deguara, R., (2005). Using Variable Length Error-Correcting Codes in MPEG-4 Video, *Proceedings of the International Symposium on Information Theory*, Adelaide, Australia, Sep. 2005

Chen, M., He, Y. & Lagendijk, R.L., (2005). A Fragile Watermark Error Detection Scheme for Wireless Video Communications, *IEEE Transactions on Multimedia*, Vol. 7, no. 2, Apr. 2005, pp. 201-211, ISBN 1520-9210

Cortes, C. & Vapnik, V., (1995). Support Vector Networks, *Machine Learning*, Vol. 20, pp. 279-297, 1995

Duda, R.O., Hart, P.E. & Stork, D.G., (2000). *Pattern Classification (Second Edition)*, Wiley-Interscience,ISBNJ 978-0-471-05669-0, New York

Ekmekci Flierl, S., Sikora, T. & Frossard, P., (2005). Coding with Temporal Layers or Multiple Descriptions for Lossy Video Transmission, *Proceedings of the International Workshop Very Low Bit-rate Video*, Sardinia, Italy, Sep. 2005

Farrugia, R.A. & Debono, C.J., (2007). Enhancing the Error Detection Capabilities of the Standard Video Decoder using Pixel Domain Dissimilarity Metrics, *Proceedings of the IEEE International EUROCON Conference*, Warsaw, Poland, Sep. 2007

Ghanbari, M. (2003). *Codec Standards: Image Compression to Advanced Video Coding*, IET, ISBN 978-0852967102, London

Girod, B. & Färber, N. (1999). Feedback-Based Error Control for Mobile Video Transmission, *Proceedings of IEEE*, Vol. 97, Oct. 1999, pp. 1707-1723, ISSN 0018-9219

Gonzalez, R.C., & Woods R.E., (2008). *Digital Image Processing*, Pearson Prentice Hall, ISBN 978-0-13-505267-9, New Jersey

Goyal, V.K. (2001). Multiple Description Coding: Compression Meets the Network, *IEEE Signal Processing Magazine*, Vol. 18, pp. 74-93, Sep. 2001

Guillemot, C. & Siohan, P., (2005). Joint Source-Channel Decoding of Variable-Length Codes with Soft Information: A Survey, *Eurasip Journal on Applied Signal Processing*, Vol. 2005, no. 6, (2005), pp. 906-927

Gupta, M.M, Jin, L. & Homma, N., (2003). *Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory*, IEEE Press, ISBN 978-0-471-21948-4, New Jersey

ITU-T Rec. P.910, (1999), *Subjective video quality assessment methods for multimedia applications*

Keerthi, S.S., Shevade, S.K., Bhattacharyya, C. & Murthy, K.R.K., (2001). Improvements to Platt's SMO Algorithm for SVM Classifier Design, *Neural Computing*, Vol. 13, no. 3, pp. 637-649, Mar. 2001

Khan, E., Lehmann, S., Gunji, H. & Ghanbari, M., (2004). Iterative Error Detection and Correction of H.263 Coded Video for Wireless Networks, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 14, no. 12, Dec. 2004, pp. 1294-1307, ISSN 1051-8215

Kumar, G., Xu, L., Mandal, M.K. & Panchanathan, S., 2006. Error Resiliency Schemes in H.264/AVC Standard, *Elsevier Journal of Visual Communications and Image Representation*, Vol. 17, no. 2, Apr. 2006, pp. 425-450

Liebl, G., Schierl, T., Wiegand, T. & Stockhammer, T., (2006). Advanced Wireless Multiuser Streaming using the Scalable Video Coding Extensions of H.264/MPEG4-AVC, *Proceedings of the IEEE International Conference on Multimedia and Expo*, Toronto, Canada, Jul. 2006

Lin, S. & Costello, D.J., (1983), *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, ISBN 9787-0132837965, New Jersey

Liu, L., Zhang, S., Ye, X. & Zhang, Y., (2005), Error Resilience Schemes of H.264/AVC for 3G Conversational Video Services, *IEEE Proceedings of International Conference on Computer and Information Technology*, Shanghai, China, 2005

Nemethova, O., Forte, G.C & Rupp, M., (2006). Robust Error Detection for H.264/AVC using Relation based Fragile Watermarking, *Proceedings of International Conference on Systems, Signals and Image Processing*, Budapest, Hungary, Sep. 2006

Nguyen, H. & Duhamel, P., (2003). Estimation of Redundancy in Compressed Image and Video data for Joint Source-Channel Decoding, *IEEE Global Telecommunications Conf.*, San Francisco, USA, Dec. 2003

Nguyen, H. & Duhamel, P., (2005). Robust Source Decoding of Variable-Length Encoded Data taking into account Source Constraints, *IEEE Transactions on Communications*, Vol. 53, no. 7, Jul. 2005, pp. 1077-104, ISSN 0090-6778

Nguyen, H., Duhamel, P., Broute, J. & Rouffet, F., (2004). Optimal VLC Sequence Decoding Exploiting Additional Video Stream Properties, in *IEEE Proceedings International Conference on Acoustic, Speech and Signal Processing*, Montreal, Canada, May 2004

Ojala, T., Pietikäinen, M. & Harwood, D., (1996). A Comparative Study on Texture Measures with Classification based on Feature Distributions, *Pattern Recognition*, Vol. 29, no. 1, Jan. 1996, pp. 51-59

Park, W. & Jeon, B., (2002). Error Detection and Recovery by Hiding Information into Video Bitstreams using Fragile Watermarking, *Proceedings of SPIE Visual Communications and Image Processing*, Vol. 4671, Jan. 2002, pp. 1-10

Platt, J.C., (1998). Fast Training of Support Vector Machines using Sequential Minimal Optimization, *Advances in Kernel Methods: Support Vector Learning*, MIT Press, ISBN 0-262-19416-3, Cambridge

Platt, J.C., (1999). Using Sparse and Analytic QP to Speed Training of Support Vector Machines, *Advances in Neural Information Processing Systems*, MIT Press, ISBN 0-262-11245-0, Cambridge

Rane, S., Baccichet, P. & Girod, B., (2006). Modeling and Optimization of a Systematic Lossy Error Protection System based on H.264/AVC Redundant Slices, *Proceedings on IEEE Picture Coding Symposium*, Beijing, China, Apr. 2006

Richardson, I.E.G (2003). *H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia*, Wiley, ISBN 0-470-84837-5, New York

Roodaki, H., Rabiee, H.R. & Ghanbari, M., (2008). Performance Enhancement of H.264 Coded by Layered Coding, *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing*, Las Vegas, USA, Apr. 2008

Rumelhart, D.E., Hiltin, G.E. & Williams, R.J., (1986). Learning Internal Representations by Error Propagation, *Parallel Processing: Explorations in the Microstructure of Cognition*, MIT Press, Vol. 1, ISBN 0-262-68053-X, Cambridge

Sabeva, G., Ben Jamaa, S., Jieffer, M. and Duhamel, P., (2006). Robust Decoding of H.264 Encoded Video Transmitted over Wireless Channels, *IEEE Workshop on Multimedia Signal Processing*, Victoria, Canada, Oct. 2006

Schwarz, H., Marpe, D. & Hannuksela, M.M., (2006). Overview of the Scalable H.264/MPEG4-AVC Extension, *Proceedings of the IEEE Conference on Image Processing*, Atlanta, USA, Oct. 2006

Specht, D.F., (1988). Probabilistic Neural Networks for Classification, Mapping or Associative Memory, *Proceedings of the IEEE Conference on Neural Networks*, San Diego, California, Jul. 1988

Stockhammer, T. & Hannuksela, M.M. (2005). H.264/AVC Video for Wireless Transmission, *IEEE Wireless Communications*, Vol. 12, no. 4, Aug. 2005, pp. 6-13, ISBN 1536-1284

Stockhammer, T., Hannuksela, M.M, & Wiegand T. (2003). H.264/AVC in Wireless Environments, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, no. 7, Jul. 2003, pp. 657-673, ISBN 1051-8215

Superiori, L., Nemethova, O. & Rupp, M., (2006). Performance of a H.264/AVC Error Detection Algorithm based on Syntax Analysis, *Proceedings of International Conference on Advances in Mobile Computing and Multimedia*, ISBN, Yogyakarta, Indonesia, Dec. 2006

Superiori, L., O. Nemethova, O. & Rupp, M., (2007). Detection of Visual Impairments in the Pixel Domain of the Corrupted H.264/AVC Packets, *IEEE Proceedings of the International Picture Coding Symposium*, Lisbon, Portugal, Nov. 2007

Tilio, T., Grangetto, M. & Olmo, G., (2008). Redundant Slice Optimal Allocation for H.264 Multiple Description Coding, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 18, no. 1, Jan. 2008, pp. 59-70, ISSN 1051-8215

Wang, G., Zhang, Q., Zhu, W. & Zhang, Y-Q. (2001), Channel-Adaptive Unequal Error Protection for Scalable Video Transmission over Wireless Channel, *Proceedings of SPIE Visual Communications Image Processing*, San Jose, USA, Jan. 2001

Wang, J.T. & Chang, P.C., (1999). Error Propagation Prevention Technique for Real-Time Video Transmission over ATM Networks, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 9, no. 3, Apr. 1999, pp. 513-523, ISBN 1051-8215

Wang,Y. & Lin, S., (2002). Error-Resilient Video Coding using Multiple Description Motion Compensation, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, no. 6, Jun. 2002, pp. 438-452, ISBN 0-7803-7025-2

Wang, Y, Reibman, A.R. & Lin, S., (2005). Multiple Description Coding for Video Delivery, *Proceedings of IEEE*, Vol. 93, no. 1, Jan. 2005, pp. 57-70, ISSN 0018-9219

Weidmann, C., Kadlec, P., Nemethova, O. and Al-Moghrabi, A., (2004). Combined Sequential Decoding and Error Concealment of H.264 Video, *IEEE Workshop on Multimedia Signal Processing*, Siena, Italy, Oct. 2004

Welzl, M., (2005). Passing Corrupted Data Across Network Layers: An Overview of Recent Development and Issues, *EURASIP Journal on Applied Signal Processing*, Vol. 2005, Jan. 2005, no. 2, 2005, pp. 242-247, ISBN 1110-8657

Wenger, S. & Horowitz, M. (2002). Flexible MB ordering – a new error resilience tool for IP-based video, *Proceedings of Tyrrhenian International Workshop on Digital Communications*, Capri, Italy, Sep. 2002

Wenger, S., (2003). H.264/AVC over IP, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, no. 7, Jul. 2003, pp. 645-656, ISBN 1051-8215

Wiegand, T. & Sullivan, G.J.,(2007). The H.264/AVC Video Coding Standard, *IEEE Signal Processing Mag*., Vol. 24, no. 2, Mar. 2007, pp. 148-153, ISBN 1053-5888

Ye, S., Lin, X. & Sun, Q., (2003). Content Based Error Detection and Concealment for Image Transmission over Wireless Channels, *Proceedings of the IEEE International Symposium on Circuits and Systems*, Bangkok, Thailand, May 2003

Zhang, C. & Xu, Y., (1999). Unequal Packet Loss Protection for Layered Video Transmission, *IEEE Transactions on Broadcasting*, vol. 45, no. 2, Jun. 1999, pp. 243-252, ISBN 0018-9316799

Zhu, C., Wang, T-K., & Hannuksela, M.M., (2006). Error Resilient Video Coding using Redundant Pictures, *Proceedings of IEEE International Conference on Image Processing*, Atlanta, GA, USA, Oct. 2006