ABSTRACT

The H.264/AVC standard employs a number of error-resilient mechanisms to correct transmission errors. These methods assume a packet-loss scenario, where all the macroblocks (MBs) contained within a corrupted slice are dropped and concealed. However, most of the MBs contained within corrupted slices provide minimal (if any) visual distortions and therefore concealing them causes a superfluous drop in the quality of the recovered video content.

This paper presents a novel error control mechanism which employs prior information and residual source-redundancy to recover the most-likelihood feasible H.264/AVC bitstream. Simulation results show that the algorithm recovers a number of corrupted sequences and achieves overall Peak Signal-to-Noise Ratio (PSNR) gains between 1dB and 2dB over the standard. The proposed solution is compatible with the H.264/AVC with no additional bandwidth requirements.

Index Terms—error control, H.264/AVC bitstream, prior information, residual source redundancy.

1. INTRODUCTION

Wireless video transmission offers a number of technical challenges [1], one of the most prominent being the extreme vulnerability of most video coding standards to transmission errors. These errors provide severe degradation in quality of the reconstructed video sequences with the effect of visually impaired macroblocks (MBs) propagating in both the spatial and the temporal domains.

H.264/AVC [2] offers several coding schemes that enhance the error resilience of the compressed video bitstream. However, these methods consider a packet-loss scenario, where the receiver discards all the corrupted packets and conceals all the MBs contained within them, even the uncorrupted ones [3]. This implies that the error resilient methods adopted by the standard operate at a lower bound since they assume a worst-case scenario.

A number of error control mechanisms are proposed in literature. The authors in [4] replaced the variable length code (VLC) tables of the MPEG-4 video compression standard with variable length error correction (VLEC) codes. However, VLEC codes reduce the compression efficiency of the codec since they have longer average codeword lengths. An analysis on the redundancy in compressed video data is found in [5], 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 proposed a Viterbi decoding method for H.263 sequences [6]. Sequential decoding with source constraints are adopted in [7], [8] to recover feasible compressed video bitstreams. However, sequential decoding algorithms introduce variable decoding delays which could be a problem in real-time applications [9].

This paper presents a novel error control mechanism which can be used to derive the most-likelihood feasible H.264/AVC sequence. This method provides an iterative solution whereby, at each symbol time, the $M$ most-likelihood sequences are stored within a list based on prior information. A set of source constraints is further adopted to eliminate the candidate solutions which provide semantically invalid H.264/AVC sequences, reducing the search space and consequently the computational complexity. This method achieves significant gains in quality when compared to the standard method with Peak Signal-to-Noise Ratio (PSNR) gains approaching 2dB.

This paper is organized as follows; the proposed error control mechanism is described in section 2 with particular emphasis on the source constrains and the prior information. Simulation results are presented in section 3 followed by final comments and conclusion in section 4.
2. PROPOSED ERROR CONTROL METHOD

Wireless video applications encapsulate each slice or frame within RTP/UDP/IP packets using the single Network Abstraction Layer Unit (NALU) mode [10]. The received payload contains a sequence having a length of \( N \) bits, whose number of VLC codewords and their original sequence are not known by the receiver and thus have to be estimated.

Once the bitstream is received, the proposed error control method creates an empty state \( S_{0,0} \) and stores it in list \( L_{k-1} \), which will contain incomplete feasible sequences. That is, sequences which satisfy all the following conditions: (1) the length of the bitstream is smaller than or equal to \( N \), (2) the number of MBs decoded by the bitstream must be smaller than or equal to the number of MBs in the slice, and (3) the decoded symbols are within the ranges allowed by the H.264/AVC standard [2]. A second list \( F \) is initialized as an empty list. This list will contain all the complete feasible VLC sequences of size \( N \) such that the number of complete MBs being decoded is equal to the number of MBs in the slice.

The proposed algorithm adopts an iterative method, where each time \( L_{k-1} \) is available, a new \( L_k \) is initialized as an empty list. For each state in list \( L_{k-1} \) the following procedure is followed, assuming that the VLC table to be loaded is known a priori:

1. Load the VLC table. Each VLC instance is made up of a codeword \( c_i \) of length \( \lambda_i \) bits with prior probability \( p_i \).
2. For each codeword in the VLC table:
   i. Extract \( \lambda_i \) bits from bitstream \( x \) and store this codeword within buffer \( b \).
   ii. Compute the distance between the two codewords using:
      \[
      m = m' + (1 - p_i) |b, c_i|
      \]  
      where \( | , | \) stands for the hamming distance between the two codewords and \( m' \) represents the metric stored in the current state.
   iii. Create a state \( S \), set the bitstream equal to the concatenation of the bitstream stored in the current state and the codeword \( c_i \), and set the new metric equal to \( m \).
   iv. Place the state \( S \) within the list \( L_k \).

At the end of each recursion, the list \( L_k \) will contain the feasible VLC sequences of \( k \) symbols. The states enclosed within the list \( L_k \) which contain complete feasible H.264/AVC sequences are stored in \( F \). On the other hand, the incomplete feasible VLC sequences are sorted in ascending order according to the metric, and the first \( M \) states are stored in \( L_{k-1} \) for the next iteration. The recursion is terminated when list \( L_k \) is empty with \( k \geq 1 \), at which point the list \( F \) will hold all the complete feasible bitstreams. The optimal sequence of codewords is then the complete feasible sequence with the smallest metric among all the sequences found in \( F \).

This error control algorithm relies on the contextual module and prior information to function appropriately. The contextual module is used during each recursion to predict the VLC table to be loaded, while the prior information is required to ensure that only the most probable partial H.264/AVC sequences are kept in memory, thus reducing unnecessary storage of data in the lists and aid convergence.

2.1. Contextual Module

Two different entropy coding methods are supported by H.264/AVC: Context-Adaptive VLC (CAVLC) and Context Adaptive Binary Arithmetic Coding (CABAC). The Baseline profile, which is generally adopted for real-time wireless applications, adopts the CAVLC entropy coding method [8], [11] and therefore is the one considered in this work. CAVLC encodes the quantized coefficients using VLC tables which are switched depending on the previous syntax elements and the VLC table adopted in the previous decoding step to achieve higher coding efficiency [11], [12].

The H.264/AVC decoding process of an MB is illustrated in Fig. 1. The decoder keeps track of the processing order of the slice it is decoding and predicts the next VLC table for the following step based on the syntax elements and the current table. The Residual block shown in Fig.1 is adopted by the Luma Residual, I16x16 DC Residual, DC Chroma Residual, and AC Chroma Residual to derive the residual coefficients. More information about these coefficients is provided in [13].

To apply the proposed error control method to H.264/AVC bitstreams, it is necessary to each time identify which VLC table must be loaded by each state in list \( L_{k-1} \). The contextual module thus stores all the relevant information for each state and is used to predict the table to be loaded next.
2.2. Prior Information

The performance of the proposed method depends on the distance measure adopted in step 2.ii. The simplest measure that can be used is the hamming distance. However, this is a hard distance measure which does not consider a priori information that is available at the decoder and thus provides limited error control capabilities for VLC sequences. This work assumes that no soft information is available and thus the metric provided in (1) simply multiples the hamming distance with the prior information.

In practice, the H.264/AVC codec employs more than 30 different VLC tables whose distributions differ from one another. The probability of occurrence of each codeword within the VLC table was derived using six different video sequences, Mobile, Coastguard, Suzie, Silent, Akiyo and Hall, encoded at QCIF resolution at 30fps. These prior probabilities were appended to each VLC table, thus making them accessible to the error control method at the decoder.

3. SIMULATION RESULTS

The proposed error control mechanism was integrated within the Jointed Model (JM) software [14] which was modified to allow the decoder to decode also damaged bitstreams. The raw video sequences were encoded at QCIF resolution at 30fps, with the format IPPP... and a data rate of 128kbps. The encoder applies a random Intra refresh rate of 5 and has 5 slices per picture. In compliance with the JM software, each slice was encapsulated within RTP/UPD/IP packets and transmitted over a Binary Symmetric Channel (BSC) which yields the highest probability of error in the slice [12], where 8 different error patterns were considered. The sequences used in this section are different from those used to derive the prior probabilities to avoid biasing of results.

The Foreman sequence was used to analyse the effect of the size $M$ on the PSNR and the results are summarized in Table I. It can be seen that if $M$ is kept too small (below 100) the proposed error control mechanism does not provide any gain in performance over the Syntax Analysis method described in [15]. This occurs mainly because the feasible list $F$ is most of the time empty and thus provides limited error control capabilities. However, significant gains in terms of PSNR are experienced as $M$ increases. This comes at the expense of an increase in the complexity of the decoder and thus $M$ cannot be too large.

Table I - PSNR for different List Size $M$

<table>
<thead>
<tr>
<th>BER</th>
<th>Syntax Analysis</th>
<th>LD $M=100$</th>
<th>LD $M=200$</th>
<th>LD $M=300$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00E-03</td>
<td>19.825</td>
<td>20.058</td>
<td>20.424</td>
<td>20.288</td>
</tr>
<tr>
<td>5.00E-04</td>
<td>22.775</td>
<td>23.266</td>
<td>23.471</td>
<td>23.649</td>
</tr>
<tr>
<td>3.00E-04</td>
<td>23.976</td>
<td>24.950</td>
<td>24.889</td>
<td>25.597</td>
</tr>
<tr>
<td>1.00E-04</td>
<td>27.941</td>
<td>27.210</td>
<td>28.338</td>
<td>29.366</td>
</tr>
<tr>
<td>5.00E-05</td>
<td>28.382</td>
<td>28.847</td>
<td>29.052</td>
<td>28.967</td>
</tr>
<tr>
<td>3.00E-05</td>
<td>30.823</td>
<td>30.407</td>
<td>31.600</td>
<td>31.765</td>
</tr>
</tbody>
</table>

The list size $M$ was set to 300 since this was found to provide the best compromise between performance and complexity. The performance of the proposed system was compared to both Syntax Analysis and the packet-loss method adopted by the standard by considering their performance on the Foreman and the Carphone sequences. These results are illustrated in Fig. 2, where the superiority of the proposed method is evident. The overall PSNR gains are between 1dB and 2dB when compared to the standard decoding method, whereas gains in the region of 1dB were achieved over the Syntax Analysis method.
The gain in quality is even more evident in the subjective results shown Fig. 3, where the Syntax Analysis method does not manage to detect a number of distorted MBs while the standard method conceals unnecessarily a number of uncorrupted MBs. Quantitatively it was found that, for this frame, the proposed decoding method achieves a PSNR gain of 5.63dB and 6.23dB over the standard decoding method and the Syntax Analysis method respectively.