Multiple-Description Video Coding Based on JPEG 2000 MQ-Coder Registers

Angelo M. Arrifano‡∗, Manuela Pereira§∗, Marc Antonini¶† and Mario M. Freire∥∗

∗IT - Networks and Multimedia Group, Dep. Computer Science, University of Beira Interior,
Rua Marquês d’Ávila e Bolama, 6201-001 Covilhã, Portugal.
‡miknix@gmail.com
§mpereira@di.ubi.pt
∥mario@di.ubi.pt
†I3S laboratory - UMR6070 - University of Nice Sophia Antipolis and CNRS
¶am@i3s.unice.fr

Abstract—Wireless channels are more prone to transmission errors than wired communications counterparts, leading to excessive packet retransmission that is very inconvenient on media streaming. The multiple-description-coding (MDC) has proven to be very effective against transmission errors, thus providing a solution for that problem.

What is presented is a method for MDC using the highly optimized and scalable JPEG 2000. Descriptions are encoded by our modified version of JPEG 2000 which produces compatible codestreams provided with key-error detection registers. The multiple-description JPEG 2000 decoder is then capable of precisely detecting transmission errors and to efficiently choose between available description-information, achieved by a clever exploitation of the EBCOT system. To test the potential of the proposed method, it is integrated as a spatial MD-Coder in a state-of-the-art joint-source channel video coder framework capable of an efficient bit-allocation between descriptions. A comprehensive set of experimental results is presented.

I. INTRODUCTION

Mobile video streaming is starting to be a common reality, specially when considering the new trend of high-definition video. However, the seamless streaming of video over wireless channels is a very challenging problem since the retransmission of corrupted packets must be avoided. This problem is difficult to circumvent, even with the increase of bandwidth in wireless channels and aggressive Quality of Service (QoS) rules.

The multiple-description-coding (MDC) is the process of encoding the source signal into several streams (descriptions) that are sent to the decoder over different network channels [1]. Each decoder (side decoder) receives its own description and produces necessary information, allowing the central decoder to optimally reconstruct the original signal. The more uncorrupted description-information is received, the better the reconstructed signal.

The MDC has proven [2] [3] to be very effective against transmission errors that affect the smoothness of video streaming. Given that, our primary motivation is the modelling of a multiple-description (MD) video coder capable of high-compression rates and low-distortion central reconstruction based on a well-established, high-performance and scalable coding system - the JPEG 2000 [4].

Due to the success of JPEG 2000, there are several proposals of MDC using JPEG 2000 in literature. In [5] the authors present a rate-distortion-based MDC compatible with JPEG 2000 that is enhanced in [6] with a MD quantization step. In [7], a prediction-compensated MDC using filter banks is presented, which is compatible with JPEG 2000. However, they do not provide a scheme for MD video coding and that is the main reason why the use of MDC makes sense. They also share a common particularity - the aim to optimize redundancy-allocation between descriptions. The proposed MDC method goes beyond that by relying on error-detection capabilities to provide an optimal central reconstruction in the absence of a full description or just a few bits.

In section II, we present the base method for MD image coding using our modified version of JPEG 2000 to encode the descriptions provided with key-registers for error-detection, which are still JPEG 2000 compatible. The multiple-description JPEG 2000 decoder is then capable to precisely detect transmission errors and efficiently choose between available description-information, achieved by a clever exploitation of the Embedded Block Coding with Optimized Truncation (EBCOT) system. In section III, the modified JPEG 2000 MD-coder is integrated as a spatial MD-coder in a state-of-the-art joint-source channel video coder framework capable of an efficient bit-allocation between descriptions [2] [3] [8]. Experimental results of this encoding system are presented in section IV.

II. MQ-CORDER REGISTRY-BASED MD IMAGE CODER

A. The MD Image Coding Scheme

The proposed MD image coding scheme is represented in fig. 1. A source signal (image) is encoded twice with the JPEG 2000 encoder presented in [9], to produce two codestreams (descriptions). The rate of each description can be adjusted using the standard JPEG 2000 (J2K) rate ($R_1$ and $R_2$ on fig. 1) control algorithm. The input rate is acquired from an
external MD-bit-allocation algorithm and not included in the MD image coding scheme.

Each description is then sent to the side decoder over the corresponding transmission channel. Our specialized side decoder version of JPEG 2000, carefully explained in section II-B, receives each description and proceeds with full decoding, producing two distinct outputs: the decoded image and the description-information. The decoded image is simply the output of the standard JPEG 2000 decoder with the description received as is. The description-information is a bitstream containing key-information about the description decoding process and includes the following information:

- MQ-Encoder registers saved during the encoding process.
- MQ-Decoder registers gathered from a side decoding process.
- Complete J2K codestream (or pointers to its location in the description).

Description-information is then read by our specialized JPEG 2000 central decoder (fig. 1), allowing it to make key decisions about the best information to use to optimally reconstruct the original image. The above explanation is based on two-channel MDC only for the sake of simplicity, the proposed method being scalable to any given set of transmission channels.

**B. The Specialized JPEG 2000 Side/Central Decoders**

Given that our JPEG 2000 encoder (fig. 1) produces codestreams provided with some special registers as described in [9], the side decoder can successfully detect which J2K segments are corrupted by comparing the encoder-saved special registry with the corresponding decoder registry, very much like explained in [9]:

The MQ-encoder is run with the RESTART [10] mode switch enabled, which forces the restart of the MQ-coder at the beginning of each coding pass. Besides the obvious error-resilience provided by this encoding mode, it is specifically needed to flush the arithmetic coder at the end of every coding pass and thus making them valid truncation points. The MQ-coder A [11] register is then saved at the end of each coding pass and later inserted into the J2K codestream. During the decoding process, the MQ-encoder-saved registers are compared with the MQ-decoder A registers at every segment. In table I we can see an example of registry checks for both side decoders using different rates $R_1$ and $R_2$. Registers are presented in the form of $[xxxx/yyyy]$, where $xxxx$ is the decoder registry and $yyyy$ is the encoder-saved registry. If they do not match, then it is highly probable that the J2K segment is corrupted [9].

When the central decoder runs, each available description-information (fig. 1) contains all the information needed to decide which description segment to use so that the reconstructed image SNR is maximized. For example, the segment status (corrupt/non-corrupt) and segment data-length.

**C. JPEG 2000 Codestream Compatibility**

Since the produced codestreams are still JPEG 2000 compliant [9], standard JPEG 2000 decoders can be used to decode a single-description, which provides similar results to the single-description coding (SDC) mode. However, the subsequent use of a side and then of a central decoder to decode a single-description still benefits from a minimum of error-detection capabilities, since corrupted segments are always discarded.

**D. MQ-coder Register Overhead**

The register overhead is 16bit [9] per segment. The total number of segments in a codestream directly influences the error-resilience level and overhead. Higher error-resilience requires a higher number of smaller segments, so a higher number of registers is used. The best error-resilience is achieved by truncating the codeblock at each coding pass (possible with

---

### Table I

<table>
<thead>
<tr>
<th>Seg.</th>
<th>Desc$_1$</th>
<th>Desc$_2$</th>
<th>Central decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>[a834/a834]</td>
<td>[a834/a834]</td>
<td>Choose d$_1$ or d$_2$</td>
</tr>
<tr>
<td>01</td>
<td>[9807/9807]</td>
<td>[9807/9807]</td>
<td>Choose d$_1$ or d$_2$</td>
</tr>
<tr>
<td>02</td>
<td>[ac02/ac02]</td>
<td>[ac02/ac02]</td>
<td>Choose d$_1$ or d$_2$</td>
</tr>
<tr>
<td>03</td>
<td>[a807/a807]</td>
<td>[a807/a807]</td>
<td>Choose d$_1$ or d$_2$</td>
</tr>
<tr>
<td>04</td>
<td>[ae93/ae93]</td>
<td>[ae93/ae93]</td>
<td>Choose d$_2$</td>
</tr>
<tr>
<td>05</td>
<td>[d006/d006]</td>
<td>[d006/d006]</td>
<td>Choose d$_1$ or d$_2$</td>
</tr>
<tr>
<td>06</td>
<td>[f806/f83fe]</td>
<td>[f83fe/f83fe]</td>
<td>Choose d$_2$</td>
</tr>
<tr>
<td>07</td>
<td>[a202/d804]</td>
<td>[d804/d804]</td>
<td>Choose d$_2$</td>
</tr>
<tr>
<td>08</td>
<td>[9002/fc04]</td>
<td>[fc04/fc04]</td>
<td>Choose d$_2$</td>
</tr>
<tr>
<td>09</td>
<td>[a202/8e04]</td>
<td>[8e04/8e04]</td>
<td>Choose d$_2$</td>
</tr>
<tr>
<td>10</td>
<td>[a802/c805]</td>
<td>[c805/c805]</td>
<td>Choose d$_2$</td>
</tr>
<tr>
<td>11</td>
<td>[e008/a404]</td>
<td>[a404/a404]</td>
<td>Choose d$_2$</td>
</tr>
<tr>
<td>12</td>
<td>[ctfe/c3ff]</td>
<td>[c3ff/c3ff]</td>
<td>Choose d$_2$</td>
</tr>
<tr>
<td>13</td>
<td>[f002/f002]</td>
<td>[f002/f002]</td>
<td>Choose d$_1$ or d$_2$</td>
</tr>
<tr>
<td>14</td>
<td>[ec02/c804]</td>
<td>[c807/c804]</td>
<td>Discard</td>
</tr>
<tr>
<td>15</td>
<td>[8000/c002]</td>
<td>[c002/c002]</td>
<td>Choose d$_2$</td>
</tr>
<tr>
<td>16</td>
<td>[900b/b802]</td>
<td>[b802/b802]</td>
<td>Choose d$_2$</td>
</tr>
<tr>
<td>17</td>
<td>[f008/ae02]</td>
<td>[ae02/ae02]</td>
<td>Discard</td>
</tr>
<tr>
<td>18</td>
<td>[8000/8000]</td>
<td>[8000/8000]</td>
<td>Choose d$_1$ or d$_2$</td>
</tr>
<tr>
<td>19</td>
<td>[9004/c004]</td>
<td>[c004/c004]</td>
<td>Discard</td>
</tr>
<tr>
<td>20</td>
<td>[8c03/ae02]</td>
<td></td>
<td>Discard</td>
</tr>
<tr>
<td>21</td>
<td>[8000/8000]</td>
<td></td>
<td>Choose d$_1$</td>
</tr>
</tbody>
</table>
the \textit{RESTART} mode switch [10]), while the lowest protection level is achieved by not truncating a codeblock at all.

Also notice that if more than two descriptions are encoded, they do not need to include MQ-encoder registers to provide error-detection capabilities to the central decoder. In fact, the central decoder can deduce which segments are corrupted by searching for the common register between all available descriptions.

III. MD VIDEO CODER BASED ON JPEG 2000

This MD video coder, described in figure 2, follows the scheme proposed in [2] and based on [3] [8], in which the product code and MAP algorithms are replaced by JPEG 2000 as presented in [8]. The use of our MQ-coder registry-based JPEG 2000 for temporal subband MD coding yields to an efficient central signal reconstruction while maintaining the excellent performance demonstrated in [8].

The video coding scheme starts by performing the temporal motion-compensated wavelet-transform (WT) presented in [8]. The MD bit-allocation (fig. 2) efficiently distributes resources to descriptions based on the redundancy parameter [3] and temporal subband rate-allocation is achieved by a rate-distortion algorithm [8] that includes an MD-bit-allocation (fig. 2) that efficiently distributes resources to descriptions based on the redundancy parameter \((R_n)\) [3]. The redundancy parameter aims to adapt redundancy to the current channel characteristics, within the flow. Afterwards, they are encoded with our MD JPEG 2000 scheme using the provided target-rate \(R\) and the bitstream is sent over a noisy channel. In the decoding process, an inverse temporal motion-compensated WT takes place and temporal subbands are decoded using the method explained in section II.

IV. EXPERIMENTAL RESULTS

We performed some experiments and compared our results with [2] that is a highly performing MDC video codec. For that purpose we used three \((2,0)\) temporal decomposition levels with 1/4 pixel motion vectors. The source sequence “foreman.cif” was encoded using a total target rate \(R_t = R_1 + R_2 = 1500\) kbps and “erik.cif” sequence using a total target rate \(R_t = R_1 + R_2 = 1300\) kbps, where only the MQ-encoder-saved register overhead is not accounted for. The Binary Symmetric Channel (BSC) and Additive White Gaussian Noise channel (AWGN) were chosen to simulate the noisy transmission channel and were applied to the J2K codestream only, leaving the video stream headers and motion vectors intact. Our spatial JPEG 2000 coder was also tuned for error-resilience, using a (irreversible) 9-7 DWT, while EBCOT codeblocks were size 8x8. Also note that we explicitly need the JPEG 2000 RESTART mode switch enabled.

Table II shows how the proposed MD Video Coder based on the central reconstruction method explained in II (REG-MDC for short) behaves according to different noise levels and description redundancy values \((R_n)\). The summarized results are the mean PSNR value of all encoded frames \((144)\). We can see that the central decoder is capable of successfully extracting non-corrupt information from the two descriptions, providing an admirable central reconstruction. Also notice that the central quality level is much higher than that of the side decoders, allowing us to conclude that the noise caused by transmission-errors was indeed corrected by our MDC. Also notice that the method is not limited to BSC, it performs well in different types of transmission channels which is the case of AWGN.

Consider the graph in figure 5 that compares the quality of our central reconstruction with [2]. Our JPEG 2000 central reconstruction method (REG-MDC) clearly outperforms the MAP algorithm at high bit-error rates while still providing similar results for low bit-error rates. In figure 5 and table II, the reason why the central reconstruction quality of erik is similar for BER \(\approx 5 \times 10^{-4}\) and BER \(\approx 4 \times 10^{-4}\) is due to the fact that the global encoding rate \(R_t = 1300\)kbps is too low, which causes some temporal subbands (chrominance

\begin{table}[h]
\centering
\caption{Side decoders VS central PSNR (decibels)}
\begin{tabular}{|c|c|c|c|}
\hline
\textbf{Foreman Sequence} \((R_t = 2000\) kbps), AWGN & & & \\
\hline
\approx BER & \(R_n\) & Side 1 & Side 2 & Central \\
\hline
54 \times 10^{-4} & 0.2 & 22.59 & 21.86 & 31.38 \\
24 \times 10^{-4} & 0.2 & 25.44 & 27.11 & 34.42 \\
6 \times 10^{-4} & 0.2 & 31.80 & 31.67 & 35.77 \\
\hline
\textbf{Foreman Sequence} \((R_t = 1500\) kbps, BSC) & & & & \\
\hline
\approx BER & \(R_n\) & Side 1 & Side 2 & Central \\
\hline
5 \times 10^{-4} & 0.2 & 27.52 & 27.69 & 33.40 \\
2 \times 10^{-4} & 0.8 & 32.55 & 31.83 & 34.11 \\
5 \times 10^{-4} & 0.8 & 28.25 & 29.63 & 33.73 \\
\hline
\textbf{Erik Sequence} \((R_t = 1300\) kbps, BSC) & & & & \\
\hline
\approx BER & \(R_n\) & Side 1 & Side 2 & Central \\
\hline
4 \times 10^{-4} & 0.2 & 27.80 & 28.65 & 30.99 \\
5 \times 10^{-4} & 0.2 & 27.35 & 28.01 & 31.00 \\
3 \times 10^{-4} & 0.3 & 28.86 & 29.17 & 31.07 \\
\hline
\end{tabular}
\end{table}
is made (only temporal) and the side decoders were only modified to output description-information to the central decoder, so their behaviour is still similar to standard JPEG 2000. Since maintaining JPEG 2000 compatibility is one of our objectives, there was no interest in improving the side decoders; therefore a direct comparison of side reconstruction with [2] is not presented. The side reconstruction quality shown in [2] outperforms the values presented in table II since they perform spatial redundancy-allocation.

**V. CONCLUSION**

In this paper we proposed a method for multiple-description coding using JPEG 2000. No other coder systems designed from ground up are required, just the well-known JPEG 2000 and a little of inexpensive exploitation of its scalable EBCOT system. The method can successfully detect transmission errors and use the available information to optimally reconstruct the corrupted signal.

The all-in-one combination of an efficient multiple-description resource-allocation, state-of-the-art motion compensation and our robust multiple-description encoding method reveals itself as very powerful, as shown in the results section.

**REFERENCES**


