

WAVELET DOMAIN DISTRIBUTED CODING FOR VIDEO

R. Bernardini, R. Rinaldo, P. Zontone
DIEGM–Università degli Studi di Udine - Italy
Via delle Scienze 208, Udine
e-mail: {bernardini, rinaldo, pamela.zontone}@uniud.it

D. Alfonso, A. Vitali
ST Microelectronics
Via C. Olivetti n. 2, 20041, Agrate Brianza - Italy
e-mail: {daniele.alfonso, andrea.vitali}@st.com

ABSTRACT

In this paper we present a wavelet domain distributed coder for video which allows for scalability and does not require any feedback channel. Efficient distributed coding is obtained by processing the wavelet transform with a suitable “folding” function and compressing the folded coefficients with a wavelet coder. At the receiver side, we use the statistical properties between similar frames to recover the compressed frame. Experimental results show that the proposed scheme has good performance when compared with similar asymmetric video compression schemes.

Index Terms— Data compression, Source coding, Video coding

1. INTRODUCTION

Distributed Source Coding (DSC) refers to the compression of multiple correlated sources that do not communicate with each other. These sources send their compressed outputs to a common decoder that performs joint decoding. A challenging problem under this paradigm, is to achieve the same efficiency as traditional coding without requiring sources to communicate with each other. Let \( \{(X_i, Y_i)\} \) be a sequence of independent and identically distributed drawings of a pair of correlated discrete random variables \( X \) and \( Y \). In [1], Slepian and Wolf showed that \( R = H(X,Y) \) is sufficient even for separate encoding of correlated sources. The counterpart of the Slepian-Wolf theorem for lossy source coding is the Wyner and Ziv’s work [2]. Recently, several schemes based on the distributed source coding principle have been proposed. In [3], Pradhan and Ramchandran presented a syndrome-based framework, successively extended to a video coding system in [4]. Other studies have been based on channel codes [5, 6, 7].

In particular, in [8], Aaron et al. apply a Wyner-Ziv coding to the pixel values of a video sequence; such an approach is extended to the transform domain in [9]. In this paper we propose a novel distributed video coder. The proposed coder works in the wavelet transform domain in order to allow for scalability and, differently from the coder in [8, 9], it does not require any feedback channel. In the proposed coder, variable \( X \) is the wavelet transform of the \( 2n \)-th frame of the original sequence and \( Y \) is the wavelet transform of the estimation of the same frame computed by the receiver on the basis of the odd frames of the original sequence. In the proposed coder, \( X \) is first mapped (via a possibly non-invertible folding function) to interval \([-M/2, M/2]\), where \( M \) is chosen on the basis of similarity between frames \( 2n \) and \( 2n-2 \). A novel aspect of the proposed scheme is that the folding function \( \phi_M \) plays the role of an “analog syndrome generator.” Since the output of the folding function is a real number, it is possible to use an efficient wavelet lossy coder to represent it. Note that spatial or other forms of scalability can be obtained as soon as a suitable coder (e.g. [10, 11]) is used.

In this paper, the reduced value \( \bar{X} \) is compressed by means of the efficient wavelet coder presented in [10]. At the receiver side, we use the statistical properties between similar frames to recover the compressed frame.

The paper is organized as follows: in Section 2 we give some preliminary remarks, in Section 3 we analyze in detail our scheme. In Section 4 we give some results and we compare the performance of our scheme to the ones described in [8, 9].

2. PRELIMINARY REMARKS

In most schemes of Wyner-Ziv type \( X \) is coded by applying Slepian-Wolf techniques to a quantized version of \( X \). In this paper, we use a different scheme which comprises three steps: 1) processing of \( X \) by means of a non-invertible folding function \( \phi_M : \mathbb{R} \rightarrow [-M/2, M/2] \) in order to obtain the reduced variable \( \bar{X} = \phi_M(X) \); 2) lossy coding of \( \bar{X} \); 3) decoding of \( X \) from quantized \( \bar{X} \) and side information \( Y \).

The two folding functions used in this paper are shown in Fig. 1. It is easy to see that if \( X, Y \) and \( M \) are such that \( |X - Y| < M/2 \), then the function in Fig. 1a (which corresponds to reduction modulo \( M \)) is a suitable folding function since \( X \) can be recovered from \( \bar{X} \) and \( Y \). An important drawback of reduction modulo \( M \) is that if \( M \) is chosen too small, errors can happen when \( X \) is reconstructed. Since the reconstructed value \( \hat{X} \) must be such that \( \hat{X} \mod M = \bar{X} \), it follows that \( X - \hat{X} \) will always be an integer multiple of \( M \) and this means the magnitude of “overflow” errors can be quite large. This drawback can be mitigated by using the folding func-
tion shown in Fig. 1b. Since the distance between the points in \( \phi_{M}^{-1}(x) \) is smaller than \( M \), overflow errors have a smaller magnitude, but are more probable. Which folding function gives the best results must be determined by experimental means.

At first, it is not entirely intuitive why separate coding of pair \((X, Y)\) should be efficient. In order to motivate our approach, we will analyze a special case with the help of a geometric interpretation and Property 1 in the following. Suppose reduction modulo \( M \) (Fig. 1a) is used and let the support of the density \( f_{x,y} \) of pair \((X, Y)\) be the shadowed parallelogram of Fig. 2. It is easy to see that the density of pair \((X, Y)\) is the dark area in Fig. 2 which corresponds to the restriction to the strip \([-M/2, M/2] \times \mathbb{R}\) of the repetition of the joint density \( f_{x,y} \).

The following result suggests that if the translated versions of the support are nicely “packed” together, this scheme can have good efficiency, at least in the high-bit rate region.

**Property 1.** Let \( A \) and \( B \) be two compact sets and let \((X, Y)\) be a pair of random variables uniformly distributed over \( C \subset A \times B \). Let \( I(X; Y) = h(X) + h(Y) - h(XY) \), where \( h(X) \) is the differential entropy of \( X \). The following inequalities hold

\[
0 \leq I(X; Y) \leq \log_2 \left( \frac{\mu(A)\mu(B)}{\mu(C)} \right)
\]

The proof is omitted for the sake of space. In our case, \( A \) is the support of \( X \), \( B \) is the support of \( Y \) and \( C \) is the support of the reduced pair \((X, Y)\). If \( X \) and \( Y \) are not uniformly distributed inside their support (or their support is not compact), \( A \) and \( B \) can be chosen as the respective typical sets. Since \( I(X; Y) \) in Property 1 is a measure of the inefficiency of coding separately \( X \) and \( Y \) (at least in the high-bit rate region [12]), Property 1 grants that variables \( X \) and \( Y \) can be separately coded without loss of efficiency when ratio \( \mu(A)\mu(B)/\mu(C) \) (i.e., the fraction of \( A \times B \) which is filled by the support of \((X, Y)\)) is close to one.

**3. PROPOSED SCHEME**

Our implemented scheme is depicted in Fig. 3. The odd frames from a video sequence are designed as Key frames and are available (uncompressed) as side information at the decoder. The even frames are the information frames.

**3.1. Encoder**

Let \( X \) be the wavelet transform of the \( n \)-th information frame. The wavelet coefficients are processed by means of a folding function \( \phi_M \) (note that both folding functions in Fig. 1 map zero in itself, preserving the zero trees in the wavelet domain). The value of \( M \) is chosen, by the Evaluation block, on the basis of the Euclidean distance \( D \) between the \( n \)-th and the \( n - 1 \)-th information frames. If \( D \) is small, one expects \( Y \) to be approximately equal to \( X \) and \( M \) can be chosen small. Otherwise \( X \) and \( Y \) will be quite different and a large value of \( M \) should be used. The experimental section describes the choice of \( M \) in more detail.

After the reduction block, the coefficients are compressed by means of an efficient wavelet coder. In our implementation the coder presented in [10] is used, but other choices (e.g., SPIHT [11]) are possible. An adaptive arithmetic encoder is used to encode the number of bits required by the coefficients.
are then coded using the efficient scheme of [10]; (II) We do not perform any folding operation (this virtually corresponds to choosing a very large $M$), and the wavelet coefficients $X$ are simply coded in intra-mode. At the receiver, we perform ML joint reconstruction from $Y$ and the quantized $\hat{X}$; (III) $\hat{X}$ is coded in intra-mode and no joint reconstruction is performed at the receiver.

As for strategy (I), if we do not consider quantization, we know that when $M > 2\|X - Y\|_{\infty} = 2\max_n|x_n - y_n|$, the correct value of each wavelet coefficient $X_n$ can be deduced from $\tilde{X}_n$ and $Y_n$. However, when we have partial knowledge of $\|X - Y\|_{\infty}$, using folding function of Fig. 1b could be instead the preferred strategy.

Fig. 4 suggests that the comparative performance of the three strategies depends strongly on the amount of motion between successive frames or, in other words, on the “quality” of the side-information. For strategy (I), we consider the cases $M > 2\|X - Y\|_{\infty}$ and $M < 2\|X - Y\|_{\infty}$. Fig. 4.a-c refer to frames with low motion (frame 120), intermediate motion (frame 76) and high motion (frame 286). It appears from the figures that, in the case of low and intermediate motion, folding is indeed beneficial. Moreover, if $M < 2\|X - Y\|_{\infty}$, the folding function of Fig. 1b gives better performance, suggesting that it is the preferred choice in case $\|X - Y\|_{\infty}$ is known approximately. For high motion (Fig. 4.c), the best results can indeed be obtained with strategy (III), but strategy (II) can be acceptable. We suggest that in a practical scheme, the quality of the side-information can be roughly estimated at the coder from the amount of motion between successive information frames.

Based on the discussion above, the proposed scheme of Fig. 3 comprises a motion evaluation block that computes the energy of the difference of successive information frames. These frames are classified into the three motion classes (very-low (VL), low-intermediate (LI), high motion frames (H)). No bits are sent for frames in the VL class, and the side-information is used for reconstruction. For frames in the LI class, symmetric folding with a fixed value $c$ (frame 76) and high motion (frame 286). It appears from the figures that, in the case of low and intermediate motion, folding is indeed beneficial. Moreover, if $M < 2\|X - Y\|_{\infty}$, the folding function of Fig. 1b gives better performance, suggesting that it is the preferred choice in case $\|X - Y\|_{\infty}$ is known approximately. For high motion (Fig. 4.c), the best results can indeed be obtained with strategy (III), but strategy (II) can be acceptable. We suggest that in a practical scheme, the quality of the side-information can be roughly estimated at the coder from the amount of motion between successive information frames.

3.2. Decoder

At the decoder side, the wavelet decoder is applied to the received bit-stream. The side information is generated by taking the two adjacent key frames of the current frame and performing temporal interpolation to get an estimate of the information frame. We have used two interpolation techniques: the first one computes the average of the pixel values at the same location from the two key frames; the second one is based on Motion Compensated (MC) interpolation with symmetric motion vectors [13]. From the knowledge of the output of the decoding process, the receiver deduces that the folded wavelet coefficient $\tilde{X}$ belongs to $I = [\tilde{X}' - \Delta/2, \tilde{X}' + \Delta/2]$, for some $X'$, and that $X$ belongs to the inverse image $\phi^{-1}_M(I)$ (see Fig. 1). $X$ is reconstructed in a maximum likelihood (ML) sense by taking the value $\hat{X} \in \phi^{-1}_M(I)$ closest to $Y$.

4. EXPERIMENTAL RESULTS

In this Section, we present some experiments to evaluate the performance of the scheme proposed in Fig. 3. Results are presented for the QCIF sequence Foreman. The wavelet transform $Y$ of the interpolated side-information is available at the decoder, while the encoder sends partial information about the wavelet transform coefficients $X$ of the information video sequence. The wavelet transform is computed using the well known 9/7 biorthogonal Daubechies filters, using a three level pyramid.

In order to evaluate the impact of the procedures described in the previous sections, we consider different strategies to code $X$. In particular: (I) $X$ is processed by one of the folding functions shown in Fig. 1. The resulting folded coefficients $\tilde{X}$
<table>
<thead>
<tr>
<th>M</th>
<th>Frame</th>
<th>PSNR (dB)</th>
<th>Rate (Kbit/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>80−nosymm</td>
<td>Frame 120</td>
<td>34</td>
<td>1200</td>
</tr>
<tr>
<td>80−symm</td>
<td>Frame 120</td>
<td>36</td>
<td>1200</td>
</tr>
<tr>
<td>110−nosymm</td>
<td>Frame 76</td>
<td>38</td>
<td>800</td>
</tr>
<tr>
<td>110−symm</td>
<td>Frame 76</td>
<td>40</td>
<td>800</td>
</tr>
<tr>
<td>no M reduction</td>
<td>no joint dec.</td>
<td>34</td>
<td>1200</td>
</tr>
<tr>
<td>no M reduction</td>
<td>no joint dec.</td>
<td>36</td>
<td>1200</td>
</tr>
</tbody>
</table>

**Fig. 4.** Performance obtained with three frames of the Foreman sequence using different values of the modular parameter M. (a) Frame 120. (b) Frame 76. (c) Frame 286.

**Fig. 5.** Rate - PSNR performance for Foreman sequence.

### 5. REFERENCES


