A Distributed Source Coding Technique for Correlated Images Using Turbo-Codes

Angelos D. Liveris, *Student Member, IEEE*, Zixiang Xiong, *Senior Member, IEEE*, and Costas N. Georghiades, *Fellow, IEEE*

***Abstract*—The Slepian–Wolf theorem states that the output of two correlated sources can be compressed to the same extent without loss, whether they communicate with each other or not, provided that decompression takes place at a joint decoder. We present a distributed source coding scheme for correlated images which uses modulo encoding of pixel values and encoding (compression) of the resulting symbols with binary and nonbinary turbo-codes, so that larger rate savings than using modulo encoding alone are achieved, practically without loss.**

***Index Terms*—Concatenated codes, distributed source coding, interleaved codes, Turbo codes.**

I. Introduction

DISTRIBUTED CODING of correlated sources is compression of correlated sources that do not communicate with each other but send their output to a common decoder. Systems with two correlated sources are presented in [1]–[4] and their theoretically achievable rates of lossless compression are proven in [1]. A practical framework for such systems was introduced in [5] based on channel coding concepts. An application of this approach to images was then suggested in [2]. Recently, more advanced schemes were proposed for binary sources [3], [6], [4].

In this letter we introduce a more efficient distributed compression technique for images than modulo encoding [2] for a specific correlation model. The two correlated sources X and Y are images whose same location pixel values have an almost Gaussian correlation, for which the modulo encoding approach exhibits a gap of at least 0.5 b/p (bits/pixel) from the theoretical limit. The reason is that there is still some remaining correlation between the same position pixel values even after both are passed through modulo encoders. Exploiting this remaining correlation using binary and nonbinary turbo-codes we come closer to the Slepian-Wolf limit by at least 0.25 b/p.

In Section II we present our correlation model and in Section III the performance of modulo encoding. In Section IV our scheme is explained in detail and Section V presents results. The conclusion sums up the paper.

II. Image Correlation Model

We consider five images (*airfield, boats, bridge, harbor, peppers*) and call them original images. Each is 512×512 and has 256 grey levels. Each time we do measurements, we generate one or more noisy versions of each image by adding white Gaussian noise in the following manner: if X is the grey level value of a pixel of one of the original images, $Y = \mathrm{round}(X + n)$ (limited in the 0–255 range) is the pixel value at the same position as X in the associated "noisy" image, where n is a random variable. In what follows, n is assumed zero-mean Gaussian with variance σ^2. The standard deviation σ and the range of n are functions of $M = 2^k$, $k < 8$, used as the modulo basis. We will limit the range of n in some cases to allow lossless encoding with modulo encoding, calling this n *truncated Gaussian noise*. The range of the truncated n extends from $-M/2 + 1$ to $M/2$ (all values with larger magnitudes are set equal to $-M/2 + 1$ or $M/2$ depending on their sign) and has standard deviation $\sigma = M/6$. When the range of n is not limited, we will take $\sigma = M/12$ so that modulo encoding (modulo M) yields practically lossless compression.

III. Source Coding With Modulo Encoding

In the distributed compression technique introduced by Ozonat [2], here referred to as *modulo encoding*, the coset idea [5] was applied on images by just taking the modulo of the pixel values with respect to $M = 2^k$, thus reducing the bits/pixel required to $k < 8$. So, if X is the grey level value of a pixel in the first image and Y is the grey level value of the pixel at the same position as X in the second correlated image, instead of transmitting both X and Y at 8 b/p without loss, X and $Y' = [Y]_M$, where $[\cdot]_M = \cdot \bmod M$, are transmitted as long as such a value of M assures lossless recovery of the second image at the decoder. This way, Y' indexes the coset that Y belongs to, i.e., $Y \in \{Y', M + Y', 2M + Y', \ldots, 256 - M + Y'\}$, and depending on the correlation model assumed, the correct decision is the coset member determined by the value of X.

When using modulo encoding with the Gaussian image correlation model, the result of the modulo operation on the noisy image pixel value is uniformly distributed. However, the actual difference between the noisy and the original pixel values is almost Gaussian; as a consequence, there is a gap between the theoretical limit and the achievable compression by modulo encoding. For the same noise variance σ^2, the truncated noise case requires k bits (modulo $M = 2^k$) and the nontruncated $(k + 1)$ bits (modulo $2M = 2^{k+1}$), whereas the actual difference $\Delta = Y - X$ is almost zero-mean Gaussian with variance $\sigma^2 = M^2/36$. The "differential entropy" of Δ is then

$$h(\Delta) \approx \frac{1}{2} \log_2(2\pi e \sigma^2) = (k - 0.5379) \text{ b}. \qquad (1)$$

$h(\Delta)$, the measured $H(Y|X)$ (entropy of the actual difference), the one-dimensional entropy of the noisy images and the entropy

Manuscript received March 26, 2002. The associate editor coordinating the review of this paper and approving it for publication was Prof. A. Haimovich. This work was presented in part at the IEEE International Conference on Acoustics, Speech and Signal Processing, Orlando, FL, May 2002.

The authors are with the Electrical Engineering Department, Texas A&M University, College Station, TX 77843 USA (e-mail: zx@ee.tamu.edu).

Publisher Item Identifier 10.1109/LCOMM.2002.803479.

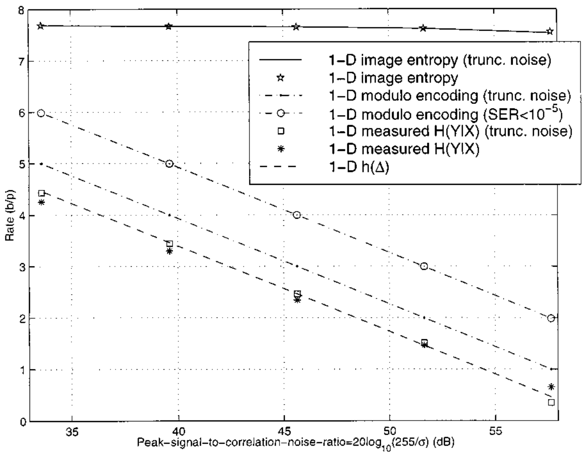

Fig. 1. Rate in b/p as a function of the peak-signal-to-correlation-noise-ratio for Gaussian noise (truncated and nontruncated).

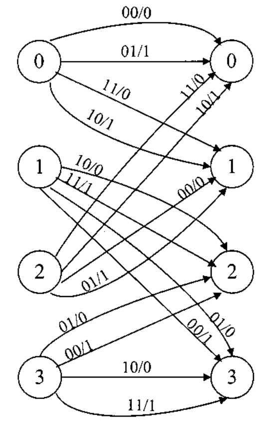

Fig. 2. Rate 1/2 encoder for binary input.

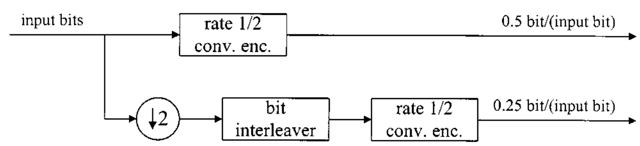

Fig. 3. Concatenated scheme of rate 3/4 (compression ratio 4/3).

of the modulo encoder output are plotted in Fig. 1 as functions of the peak-signal-to-correlation-noise-ratio. Modulo encoding falls short by about 0.5 b/p in the truncated Gaussian noise case and by about 1.5 b/p in the nontruncated case. In the latter case there is a small loss which is expressed in terms of symbol (pixel) error rate (SER). The SER in all the modulo encoding points with nontruncated noise in Fig. 1 is less than $4 \cdot 10^{-6}$, i.e., on average at most one pixel is modified in a whole image.

IV. Source Coding Using Turbo-Codes

If only the interframe correlation is taken into account, the gap between the optimum rate and the one achieved with modulo encoding can be made smaller with powerful channel coding. To better understand the suggested improvement, we can equivalently view the system after the modulo operation as having two sources with the M-ary correlated outputs $X' = [X]_M$ and $Y' = [Y]_M$ and X' being available without loss at the joint decoder. The loss due to channel coding should be so small that the overall SER in the truncated case is negligible and in the nontruncated case is not increased noticeably. Therefore, we will require that the additional SER introduced by the channel coding, i.e., $\Pr[\hat{Y}' \neq Y']$, is less than 10^{-6}.

The codes we will use are concatenated interleaved trellis codes, resembling the structure of turbo-codes, as good performance results have already been reported when applying them to the Slepian–Wolf problem [3], [6], [4]. However, the issue of analyzing and designing concatenated codes for such an application is still open and as in this scenario concatenated recursive encoders are not straightforward to apply, we will only try to maximize the minimum Hamming or Euclidean distance of the component codes, more specifically of the principal trellis of them [5]. Furthermore, the codes of [3], [6], [4] are restricted to binary correlated sources and thus cannot lead to noticeable savings at low error rates, with the exception of the highest correlation case ($\sigma = 1/3$ with truncated noise).

For the case $\sigma = 1/3$ with truncated noise, we used a similar scheme as in [3], [6], [4] to get back at least half of the loss, i.e., 0.25 b/p, as shown in the simulations in the next section. Following a different code design approach from [3], [6], [4], though using more conventional code structures as in [5], we employ the 4-state rate-$1/2$ $[1+D+D^2, 1+D^2]$ convolutional code modifying it to form the encoding trellis of Fig. 2. Since the decoder has perfect knowledge of the output bit, it always uses one of the two parallel transitions, i.e., it is like using two trellises, a principal and a complementary one [5]. The decoding is done with the BCJR algorithm as in [7], where the original image pixel values modulo-2 are used as distorted versions of the associated noisy image pixel values LSBs. The statistics of this distortion, i.e., the probability of a bit being flipped, are assumed to be known at the decoder.

The concatenated encoder is shown in Fig. 3. To achieve very low bit error rate ($< 10^{-6}$), the code of Fig. 2 is first used to encode the LSB of all the pixel values in two image rows forming one codeword of length 512. Then we take the 512 LSB's of every other pixel value of the same two rows, interleave and encode them with the same code getting 256 more output bits. So the true compression ratio is $1024/768 = 4/3$, or each pixel value is represented by 0.75 b. In the decoding process, extrinsic information is exchanged only for the common bits of the two codes (512 bits) following [7].

In the rest of the cases with smaller correlation between images, nonbinary codes can better exploit the noise statistics. These codes operate directly on the nonbinary modulos of the pixel values $X' = [X]_M$ and $Y' = [Y]_M$ of the two images ($M = 2^k, k > 1$). Following Ungerboeck's design rules [8], the component encoder used for the general case with rate $(km - 1)$ bits per m input symbols (each symbol $\in \{0, 1, \ldots, M\}$ and $m = 1, 2, \ldots$), is shown in Fig. 4 for $m = 2$. The output values are shown in the form of an $(M/2)$-ary symbol followed by $(m-1)$ M-ary symbols. For $m = 1$, only the first input symbol and the first output $(M/2)$-ary symbol are taken ($M/2$ parallel transitions). For $m \geq 2$, there are $M^m/2$ parallel transitions for all possible $M^m/2$ combinations of values of the first, second and so on symbols, with m input symbols and 1 $(M/2)$-ary output symbol and $(m-1)$ M-ary output symbols per transition. The ith input symbol ($i > 1$) is related to the $(i-1)$th M-ary output symbol exactly as the second input symbol is related to the single M-ary output symbol (function of x) in the $m = 2$ case.

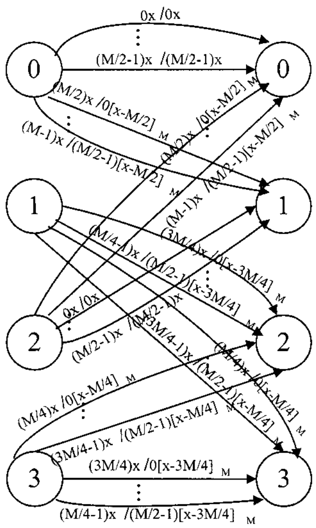

Fig. 4. Rate $(km-1)/m$ encoder for M-ary input ($(km-1)$ bits per m symbols, $M = 2^k \geq 4$): $m = 2$, $x \in \{0, 1, \ldots, M-1\}$ and $[\cdot]_M = \cdot\,\mathrm{mod}M$. For a fixed x, there are $(M/2)$ parallel transitions connecting a pair of states for $(M/2)$ successive values of the first input symbol (0 follows $M-1$).

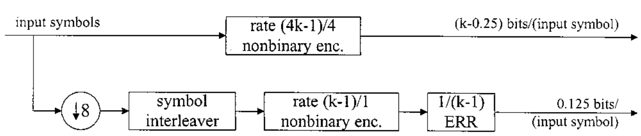

Fig. 5. Concatenated coding scheme of overall rate $(k-0.125)/k$ in output bits per input bit, i.e., $(k-0.125)$ b/p (ERR = excess redundancy remover).

The concatenated scheme of Fig. 5 recovers 0.125 b/p for all $M \geq 4$. The excess redundancy remover (ERR) removes the information of the output of the second encoder that is included in the output of the first encoder corresponding to the same input symbol. The downsampling picks every other symbol from those encoded with $(k-1)$ bits in the first code. Let a denote such a M-ary symbol and c and c' denote its encoded $M/2$-ary value from the first and second code, respectively. From Fig. 4 one can see that $[a-c]_M$ and $[a-c']_M$ are integer multiples of $M/4$ and so $[a]_{M/4} = [c]_{M/4} = [c']_{M/4}$. But since $c' \in \{0, 1, \ldots, M/2-1\}$, it suffices for the ERR whose input is c' to give the single bit information on whether $c' \geq M/4$ holds or not.

The decoder of the scheme of Fig. 5 consists of the inverse ERR module and two nonbinary BCJR modules operating as in [9]; extrinsic information is exchanged only for the common symbols of the two codes (1/8 of all symbols).

V. Simulation Results

We apply the above to the five images to measure the SER and the bit-rate savings. To ensure that the SER is below 10^{-6} in the truncated noise case and below 10^{-5} in the nontruncated case, each simulation point was run for 20 noisy realizations of each of the five images. In both cases, the probability of having a modulo-M symbol in error was measured to be less than 10^{-6}, as after 8 iterations there were no more such errors for all results in Fig. 6. Hence, the overall SER is below 10^{-6} for the truncated noise case and the overall SER does not increase noticeably for the nontruncated noise case, i.e., it is still less than $5 \cdot 10^{-6}$.

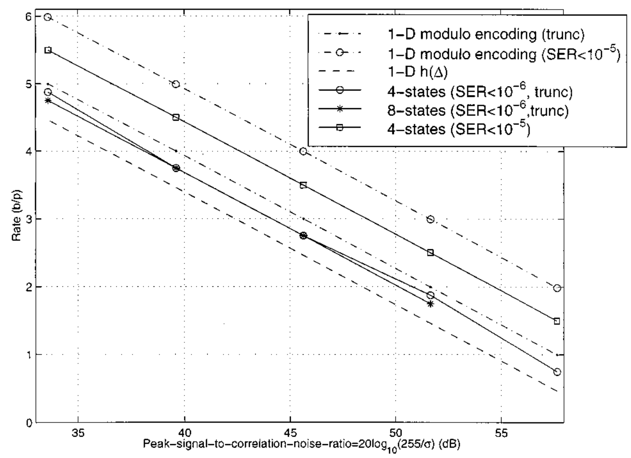

Fig. 6. The curves of Fig. 1 with the simulation results for both the truncated and nontruncated noise cases.

All simulations of Fig. 6 use the same s-random interleaver of length 512 permuting M-ary symbols. The 4-state cases refer to the code designs of Section IV. More complex coding helps gain even more, as the 8-states simulated points show, which with similar structure as in Figs. 4 and 5 and following the rules of [8], lead to slightly larger rate savings for some of the nonbinary truncated noise cases. The concatenated encoder saving 0.25 b/p uses a rate-$(2k-1)/2$ code (Fig. 4) first, downsampling by 4, interleaving and a rate-$(k-1)/1$ code followed by a rate-$1/(k-1)$ ERR. For the nontruncated noise cases, both component codes have similar structure as that of Fig. 4 with rate-$(k-1)/1$. The second encoder is preceded by a downsampler by 2 and interleaving and followed by a rate-$1/(k-1)$ ERR. This concatenated encoder saves 0.5 b/p.

VI. Conclusion

We proposed a distributed source coding technique for correlated still images based on turbo-codes. Simulation results showed that we can come as close as 0.25 b/p to the theoretically predicted limits of the Slepian–Wolf theorem for this application.

References

[1] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 471–480, July 1973.
[2] K. Ozonat, "Lossless distributed source coding for highly correlated still images," Electrical Eng. Dept., Stanford Univ., Stanford, CA, Tech. Rep., 2000.
[3] J. Bajcsy and P. Mitran, "Coding for the Slepian-Wolf problem with turbo codes," in *Proc. IEEE Globecom*, 2001, pp. 1400–1404.
[4] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proc. IEEE Data Compression Conf.*, 2002, pp. 252–261.
[5] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," in *Proc. IEEE Data Compression Conf.*, 1999, pp. 158–167.
[6] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Commun. Lett.*, vol. 5, pp. 417–419, Oct. 2001.
[7] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A soft-input soft-output APP module for iterative decoding of concatenated codes," *IEEE Commun. Lett.*, vol. 1, pp. 22–24, Jan. 1997.
[8] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55–67, Jan. 1982.
[9] J. Berkmann, "On turbo decoding of nonbinary codes," *IEEE Commun. Lett.*, vol. 2, pp. 94–96, Apr. 1998.