ERROR CORRECTION BY MEANS OF ARITHMETIC CODES: AN APPLICATION TO RESILIENT IMAGE TRANSMISSION

M. Grangetto, G. Olmo

CERCOM, Center for Multimedia
Radio Communications
Politecnico di Torino
Dipartimento di Elettronica
Corso Duca degli Abruzzi 24
10129 Torino - Italy
E-mail: `(grangetto,olmo)@polito.it`

P. Cosman

University of California, San Diego,
Dept. of Electrical and
Computer Engineering,
9500 Gilman Drive, La Jolla,
CA 92093-0407, USA
E-mail: `pcosman@code.ucsd.edu`

ABSTRACT

In this paper, two novel *maximum a posteriori* (MAP) estimators for the decoding of arithmetic codes in the presence of transmission errors are presented. Trellis search techniques and a forbidden symbol are employed to obtain forward error correction. The proposed system is applied to lossless image compression and transmission across the BSC; the results are compared in terms of both performance and complexity with a traditional separated source and channel coding approach based on convolutional codes.

1. INTRODUCTION

Universal access to multimedia data is one of the major objectives of emerging communication systems. The extension of services offered to mobile users, from traditional voice traffic to complex data sources such as web-pages, images, video and music along with the constraints imposed by the tetherless environment are boosting a considerable amount of research in the field of wireless multimedia communications. In particular, many interdisciplinary solutions are being investigated, and the interfaces among system layers are becoming richer in information content [1]. This approach is noticeable in a number of traditionally separated fields: novel compression standards, such as JPEG2000 [2] for still images and H.26L [3] for video sequences, are improving compression efficiency, but at the same time they devote great attention to transmission and error resilience; conversely CDMA2000 and UMTS systems are being designed with multimedia traffic in mind.

This scenario is generating a great interest in the development of novel and efficient *Joint Source/Channel Coding* (JSCC) techniques. Channel bandwidth and power constraints, delay limitations and error protection required by the application are emphasizing the practical shortages of Shannon's source-channel separation theorem [4], when applied to mobile multimedia communications. JSCC techniques are founded on the fact that in practical cases the source encoder is not able to exactly decorrelate the input sequence; some implicit redundancy is still present in the compressed stream and can be properly exploited by the decoder for error control. As a consequence, it is possible to improve the decoder performance by considering source and channel coding jointly.

Much research effort has been devoted to the joint source channel decoding of variable length codes, in particular *Huffman codes* [5, 6, 7]. In [5, 6], the residual redundancy in the source encoder output is represented with a Markov model, and is used as a form of implicit channel protection at the decoder side; exact and approximate maximum *a posteriori* (MAP) sequence estimators are proposed. Results are provided in the case of image transmission across the binary symmetric channel (BSC). In [7], soft MAP source decoding is investigated and applied to the transmission of MPEG-4 bitstreams.

In this paper, the JSCC approach is applied to *Arithmetic Coding* (AC). AC is nowadays the most powerful entropy coding tool [8], and is replacing Huffman coding in novel standards such as JPEG2000 and H.26L. On the other hand, AC is extremely fragile in the presence of transmission errors; unlike Huffman codes, AC has poor resynchronization capability, and a single bit error in the compressed stream can propagate all along the compressed block. Moreover, the residual redundancy in the compressed stream is usually negligible, preventing any MAP decoding attempt. Nevertheless, it is possible to perturb the arithmetic coder source model in order to keep some residual redundancy at the expense of compression efficiency. This idea was first introduced by Boyd *et al.* in [9] and extended in [10, 11] to provide continuous error detection during arithmetic decoding. The presence of known residual redundancy can be exploited for error correction as well. Some preliminary work can be found in [12], where the error correction is performed in the case of transmission over an AWGN channel; binary signalling with *null zone* soft decoding is employed. The performance is evaluated in terms of packet recovery rate for differentially encoded images. In [7], a JSCC concatenated scheme based on AC and trellis coded modulation is presented and applied to image transmission as well.

In this paper, we address the problem of MAP decoding of AC in the presence of transmission errors; AC with a forbidden symbol and trellis search techniques are used. The decoding task is formulated in terms of classical MAP estimation. Some preliminary research by the same authors was presented in [13] for lossy image compression. In the present paper, novel results are presented in the case of lossless compression of grayscale images and transmission across the BSC. The performance is compared with a standard separated source and channel scheme based on conven-

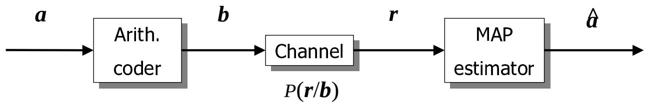

Fig. 1. Transmission system block diagram.

tional AC and rate compatible convolutional codes.

2. ARITHMETIC CODING WITH A FORBIDDEN SYMBOL

The objective of AC is to map a sequence of input symbols $\mathbf{a}$ onto a binary string $\mathbf{b}$ representing the probability of the input sequence. This is accomplished according to the available source model, and the compression efficiency mainly depends on the accuracy of the model.

In this paper, we consider the particular case of the binary memoryless source; L bits generated by grouping L outcomes of the binary memoryless source constitute the fixed length frame to be encoded, i.e., $\mathbf{a} \in \{\mathbf{a}_i\}_{i=1}^{2^L}$. This source is fully described by the probability of the binary symbols, P_0 and $P_1 = 1 - P_0$ respectively.

The AC encoding is an iterative task, performed by progressively refining the probability interval to which the input frame $\mathbf{a}$ belongs. The output sequence $\mathbf{b}$ corresponds to the shortest binary string, which represents a binary value contained in the interval; decoding follows the dual process. It is worth remarking that both encoding and decoding can be performed sequentially, by applying interval normalization strategies. The recursive decoding task is extremely sensitive to bit errors. Even a single flipped bit in the output string can cause irreversible desynchronization. Paradoxically, it is this poor resynchronization ability that allows powerful continuous error detection. In [9, 10], a *Forbidden Symbol* (FS) μ with probability $P_\mu = \epsilon$ is introduced in the input alphabet, but it is never transmitted. The introduction of the FS implies a perturbation of the source model by a factor $1 - \epsilon$, thus reducing compression efficiency. The rate redundancy amounts to $R_x = -\log_2(1-\epsilon)$ bits/symbol [10]. If the FS is decoded, this means that transmission errors have occurred. In [10], it is shown that the probability that the number of erroneously decoded bits before an error is detected is greater than n is $(1-\epsilon)^n$. Therefore, a large value of ϵ assures fast error detection and easies the correction task, but it greatly reduces the compression efficiency. On the contrary, a small value of ϵ does not impact compression efficiency but there will be a large error detection delay.

3. PROPOSED MAP DECODING SCHEME

The introduction of the FS provides a simple and robust means to obtain continuous error detection [11]. Moreover, the coding redundancy can be used by a MAP decoder in order to attempt error correction as well. The transmission system considered in this paper is shown in Fig. 1, where the input frame $\mathbf{a} \in \{\mathbf{a}_i\}_{i=1}^{2^L}$ is encoded by means of AC, using a FS with probability ϵ. An *End Of Frame* (EOF) symbol is used to terminate each frame (the role of EOF is crucial for the decoding algorithm and it will be analyzed in the following). The input sequence $\mathbf{a}$ is mapped onto the binary string $\mathbf{b} \in \{\mathbf{b}_i\}_{i=1}^{2^L}$ of variable length N bits, and transmitted across the channel with transition probability $P(\mathbf{r}/\mathbf{b})$.

The received sequence $\mathbf{r}$, possibly affected by errors, is processed by the MAP estimator that selects the most probable sequence $\hat{\mathbf{a}} : P(\hat{\mathbf{a}} = \mathbf{a}_k/\mathbf{r}) \geq P(\mathbf{a}_i/\mathbf{r})\ \forall i \neq k$. $P(\mathbf{a}_i/\mathbf{r})$ represents the so called decoding metric, and can be expressed as

$$P(\mathbf{a}_i/\mathbf{r}) = \frac{P(\mathbf{r}/\mathbf{a}_i)P(\mathbf{a}_i)}{P(\mathbf{r})} = \frac{P(\mathbf{r}/\mathbf{b}_i)P(\mathbf{a}_i)}{P(\mathbf{r})} \qquad (1)$$

$P(\mathbf{a}_i)$ represents the a priori probability of transmitting the string $\mathbf{a}_i$. On the other hand, the term $P(\mathbf{r})$ cannot be easily evaluated and in the following it will be approximated by 2^{-N}, where N is the length in bits of the received string. This approximation assumes that all the received sequences of equal length are equally likely; the assumption is not satisfied by variable length AC; however the exact evaluation of this term would require as much effort as the MAP decoding itself and it is not feasible in practice. This simplification is proposed also in [5, 6] in the case of Huffman codes and it provides satisfactory results.

The most direct approach to MAP decoding should be the evaluation of metric (1) for the subset $\mathbf{B}_N$, containing the codewords $\mathbf{b}_i$ of length N. However, for reasonable input string length L, the exhaustive approach is infeasible, and it is essential to resort to a suboptimal criterion in order to reduce the search space dimension. A number of techniques for visiting trees and trellises have been proposed in the past, the most popular one being the Viterbi algorithm; a complete survey can be found in [14]. These techniques usually require i) a trellis representation of the search space and ii) an additive branch metric. We can recast the search for the best $\mathbf{b}_i$ as a search among all possible binary strings of length N $\{\mathbf{x}_i\}_{i=1}^{2^N} \supseteq \mathbf{B}_N$. The $\{\mathbf{x}_i\}_{i=1}^{2^N}$ can be represented by a tree that grows exponentially with N. The metric (1), in its logarithmic form, can be easily decomposed into additive branch terms. The channel term is computed comparing the received $\mathbf{r}$ sequence and the explored branch. The source term is obtained attempting partial arithmetic decoding of a given tree path; in the case $\mathbf{x}_i \notin \mathbf{B}_N$, the FS will be revealed with a certain delay, and the explored path will be pruned.

For tree exploration, we tested two well known techniques known as *stack algorithm* (SA) and the *M-algorithm* (MA) [14]. SA is a *metric first* technique: the best path selection is based on a greedy approach, extending at each iteration the best stored path, i.e., the one with the best accumulated metric (1). This is accomplished by storing all the visited paths in an ordered list, with maximum length M. Each element of the list contains the accumulated metric and the state information for sequential arithmetic decoding. At each iteration, the best stored path is extended one branch forward. The extended path is dropped if the FS is revealed or if the number of decoded bits exceeds L. The branching goes on until the stopping criterion is fulfilled. In our implementation, the decoding procedure stops when the best path in storage corresponds to a valid input sequence $\mathbf{a}_i$ of length L, terminated by the EOF symbol. The decoding of EOF is crucial in order to guarantee a correct termination of the tree. The exploration of the last leaves of the tree, corresponding to the last bits in the received sequence, cannot rely on FS detection because of the decoding delay. MA limits the search space to the M best paths at each depth of the tree and is classified as a *breadth first* technique. At each iteration, all the stored paths, which are characterized by the same depth n

Table 1. SA and MA performance as a function of the memory constraint M. BER, percentage of decoding failures P_F and average frame decoding time ΔT are reported. BSC transition probability is $p_b = 5 \cdot 10^{-3}$ and $\epsilon = 0.2$.

M	BER	FER	P_F	ΔT
SA				
64	5.7 10^{-4}	1.1 10^{-1}	9.6	8.2 ms
128	1.4 10^{-4}	3.3 10^{-2}	2	10 ms
256	10^{-4}	1.6 10^{-2}	0.6	14 ms
512	4.7 10^{-5}	1.1 10^{-2}	0.1	18 ms
1024	3.2 10^{-5}	10^{-2}	0	22 ms
2048	1.7 10^{-5}	9.5 10^{-3}	0	24 ms
MA				
64	3.7 10^{-5}	1.3 10^{-2}	0.4	276 ms
128	5.4 10^{-5}	10^{-2}	0.1	807 ms
256	1.9 10^{-5}	8.5 10^{-3}	0	2.7 s
512	1.7 10^{-5}	8.9 10^{-3}	0	9.6 s

in the tree, are extended one step forward; the same dropping rules of the SA are then applied and only the M best paths at depth $n+1$ are stored. When the algorithm reaches the maximum depth $n = N$, the best stored path, terminating with EOF, is taken as the best estimate $\hat{\mathbf{a}}$. As with the Viterbi algorithm, both SA and MA allow sequential decoding, since there is a similar merging effect of all paths after a certain delay of D received bits [14].

4. RESULTS

The proposed method has been tested employing a simple lossless image compression scheme based on a zero order predictor, proposed in JPEG lossless coding system. The predicted pixel at row i and column j is $\hat{x}_{i,j} = x_{i,j-1} + x_{i-1,j} - x_{i-1,j-1}$. In the case of 256 levels grayscale images the prediction error is mapped on 9 bits symbols; frames of 256 symbols ($L = 2304$ bits) are formed and encoded by means of AC with FS; each frame is terminated with an EOF symbol with probability 0.05, which has been selected as a tradeoff between rate overhead and error detection capability. The resulting variable length packets are then transmitted over the BSC. The packet lengths along with the a priori probability P_0 are sent as side information to the decoder, adding to each packet a header, protected using a (3,1,4) convolutional code. The proposed scheme achieves a coding rate of 5.1 bits per pixel (bpp) on the *GIRL* 256×256 test image when no FS is used. (Note that the optimization of the compression performance is beyond the scope of this paper.)

First we investigated the dependence of the SA and MA performance on the memory parameter M. We evaluated Bit Error Rate (BER), Frame Error Rate (FER), percentage of decoding failures P_F and average frame decoding time ΔT obtained by an Intel Pentium 4 (512 MB RAM). BER is computed on source bits, and frames whose decoding has failed are not included in its computation. We ran simulations on 100000 frames, employing the *GIRL* test image. Results are summarized in Tab. 1 for a BSC crossover probability $p_b = 5 \cdot 10^{-3}$ and $\epsilon = 0.2$. It is worth noticing that the two algorithms exhibit rather different behavior. The SA greedy approach requires a large amount of memory ($M = 2048$) for optimal performance, but the average decoding time is always very limited. In fact, in SA case, the average number of visited paths in the tree depends on the values of P_0 and p_b and not only on M. On the contrary, MA visits approximately the same number of paths, given M, thus exhibiting a constant decoding time independent of the a priori conditions. The value of M, required for optimal performance, is smaller than the one in SA, but the decoding complexity becomes soon quite prohibitive. On the other hand, when the memory must be kept limited, e.g., case $M = 64$, MA appears more robust than SA. These considerations help in selecting the best value of M according to system constraints. In the following, we employ SA and MA with memory parameters $M = 2048$ and $M = 256$ respectively, thus keeping complexity at reasonable levels without significantly impairing the performance.

Table 2. Decoded BER and FER, average number of bit errors per frame K, percentage of decoding failures P_F and average frame decoding time ΔT for SA with $M = 2048$, MA with $M = 256$ and RCPC. $p_b = 10^{-3}$ and 10^{-2}. The RCPC coding rate R_C, the coding rate R expressed in bpp, and the value of ϵ are reported as well.

$p_b = 10^{-3}$						
SA ($M = 2048$)						
ϵ	R (bpp)	BER	FER	K	P_F	ΔT
0.05	5.67	5.3 10^{-5}	5.2 10^{-3}	27	0.1	46 ms
0.1	6.32	5.4 10^{-6}	1.8 10^{-3}	6	0	5 ms
MA ($M = 256$)						
ϵ	R (bpp)	BER	FER	K	P_F	ΔT
0.05	5.67	2.2 10^{-4}	5.9 10^{-3}	84	0	2.3 s
0.1	6.32	8.6 10^{-6}	1.5 10^{-3}	12	0	2.4 s
RCPC						
R_C	R (bpp)	BER	FER	K	P_F	ΔT
8/9	5.67	3.1 10^{-3}	2.6 10^{-2}	246	0	25 ms
4/5	6.32	4.1 10^{-4}	3 10^{-3}	283	0	26 ms
$p_b = 10^{-2}$						
SA ($M = 2048$)						
ϵ	R (bpp)	BER	FER	K	P_F	ΔT
0.05	5.67	2.5 10^{-2}	6.6 10^{-1}	164	51.3	15 s
0.1	6.32	4.5 10^{-3}	1.8 10^{-1}	91	8.7	4.2 s
0.2	7.59	1.1 10^{-4}	1.7 10^{-2}	17	0.4	250 ms
MA ($M = 256$)						
ϵ	R (bpp)	BER	FER	K	P_F	ΔT
0.05	5.67	8.2 10^{-2}	6.5 10^{-1}	342	31.6	2.3 s
0.1	6.32	1.3 10^{-2}	2.2 10^{-1}	178	8.6	2.5 s
0.2	7.59	6.6 10^{-5}	1.7 10^{-2}	12	0.5	2.7 s
RCPC						
R_C	R (bpp)	BER	FER	K	P_F	ΔT
8/9	5.67	1.6 10^{-1}	9.3 10^{-1}	342	0	22 ms
4/5	6.32	7 10^{-2}	4.5 10^{-1}	319	0	26 ms
2/3	7.59	2.2 10^{-3}	1.6 10^{-2}	273	0	28 ms

Next, we want to validate the SA and MA performance by comparing them with a classical separated approach where each arithmetic coded frame is protected by means of RCPC with coding rate R_C; no error detection tools are embedded in AC and the RCPC codes proposed in [15], with memory 6 and punctured rate $1/3$ are used. In Tab. 2 we report results for $p_b = 10^{-3}$ and

Table 3. Packet recovery rates (%) obtained by proposed algorithms, depth first (a) and breadth first (b) decoding techniques in [12].

ϵ	[12]-a	[12]-b	SA	MA
$p_b = 10^{-3}$				
0.08	96.72	99.99	99.72	99.77
0.16	99.17	99.99	99.87	99.89
$p_b = 10^{-2}$				
0.08	0.39	38.53	55.90	66.10
0.16	17.04	92.03	96.39	96.11

$p_b = 10^{-2}$ and different coding rates. Besides BER, FER and P_F, the average number of bit errors per frame K is reported. In all situations, JSCC decoders exhibit an improvement of one order of magnitude in terms of BER. In fact the joint approach prevents error propagation that takes place in the separated RCPC case, where a single convolutional decoding error can be propagated all along the frame by conventional arithmetic decoding. The advantage of MAP decoders is further witnessed by the average number of bit errors per frame; RCPC decoder always presents very high values of K if compared to those obtained through MAP algorithms, even at reduced coding rates. Moreover SA and MA yield better or comparable results in terms of FER; it is important to keep in mind that, in the presence of an external frame error detection mechanism, FER represents the only system performance measure. It is worth pointing out that SA generally performs slightly better than MA in terms of both BER and FER. Finally it is noticeable that SA with $M = 2048$ assures excellent performance with an attractive decoding time in the case $p_b = 10^{-3}$. SA complexity with a large value of M appears indeed quite prohibitive in the case of high $p_b = 10^{-2}$. On the contrary, MA decoding time is always constant and it depends only on the value of M. It is important to notice that both the proposed MAP estimators can fail the decoding in adverse channel conditions ($p_b = 10^{-2}$). In some multimedia applications, a failure declaration does not necessarily represent a problem and it could be better than forwarding erroneous data to a complex and error sensitive decoder.

Finally, in Tab. 3 the performance is compared with results available in the literature [12], in terms of Packet Recovery Rates (PRR). In [12] Pettijohn *et al.* propose a metric first (a) and a depth first (b) decoder for AC with FS. The simulation conditions are similar to those used in this paper except for the test image and some implementation details involving side information and frame organization. Nevertheless the most significant differences reside in the decoding algorithms: the authors in [12] prune the search tree using soft information and they employ the Euclidean distance as decoding metric. It is worth noticing that the proposed MAP JSCC decoders exhibit similar performance in the case $p_b = 10^{-3}$, and they show a considerable improvement in adverse channel conditions. We must recall that the proposed MAP decoders are able to provide good results even without the exploitation of any soft information.

5. CONCLUSIONS

AC with FS and sequential decoding techniques have allowed us to design an error resilient entropy coder, able to guarantee superior performance compared to a traditional separated source and channel coding approach. The proposed technique yields significant improvement in terms of BER and FER, while limiting the complexity at a reasonable level. Soft decoding and iterative systems, along with the extension of the proposed algorithms to adaptive models are the main objectives of our future research in the field.

6. REFERENCES

[1] J. D. Gibson, *Multimedia communications: direction and innovations*, Academic Press, San Diego, 2001.

[2] *JPEG2000 image compression standard*, docs. ISO/IEC FCD 15444-1, 1999.

[3] T. Wiegand, *JVT working draft 2*, JVT-B118r7, Apr. 2002.

[4] S. Vembu, S. Verdu, and Y. Steinberg, “The source channel theorem revisited,” *IEEE Trans. on Inform. Theory*, vol. 41, pp. 44–54, Jan. 1995.

[5] K. Sayood, H. Otu, and N. Demir, “Joint source/channel coding for variable length codes,” *IEEE Trans. Commun.*, vol. 48, no. 5, pp. 787–794, May 2000.

[6] M. Park and D. Miller, “Joint source-channel decoding for variable-length encoded data by exact and approximate MAP source estimation,” *IEEE Trans. Commun.*, vol. 48, no. 1, pp. 1–6, Jan. 2000.

[7] M. Bystrom, S. Kaiser, and A. Kopansky, “Soft source decoding with applications,” *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 11, no. 10, pp. 1108–1120, Oct. 2001.

[8] I. H. Witten, J. G. Cleary, and R. Neal, “Arithmetic coding for data compression,” *Commun. ACM*, , no. 6, pp. 520–540, June 1987.

[9] C. Boyd, J. Cleary, S. Irvine, I. Rinsma-Melchert, and I. Witten, “Integrating error detection into arithmetic coding,” *IEEE Trans. Commun.*, vol. 45, no. 1, pp. 1–3, Jan. 1997.

[10] J. Chou and K. Ramchandran, “Arithmetic coding-based continuous error detection for efficient ARQ-based image transmission,” *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 861–867, June 2000.

[11] R. Anand, K. Ramchandran, and I. V. Kozintsev, “Continuous error detection (CED) for reliable communication,” *IEEE Trans. Commun.*, vol. 49, no. 9, pp. 1540–1549, Sept. 2001.

[12] B.D. Pettijohn, M.W. Hoffman, and K. Sayood, “Joint source/channel coding using arithmetic codes,” *IEEE Trans. Commun.*, vol. 49, no. 9, pp. 1540–1548, Sept. 2001.

[13] M. Grangetto and P. Cosman, “MAP decoding of arithmetic codes with a forbidden symbol,” in *Proc. of ACIVS 2002*, Ghent, Belgium, Sept. 2002.

[14] J. B. Anderson and S. Mohan, *Source and channel coding*, KLUWER, 1991.

[15] J. Hagenauer, “Rate-compatible punctured convolutional codes (RCPC codes) and their applications,” *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1988.