A factor graph framework for joint source-channel decoding of images.*

Igor Kozintsev, Ralf Koetter and Kannan Ramchandran
University of Illinois at Urbana-Champaign, UC Berkeley
igor@ifp.uiuc.edu, koetter@uiuc.edu, kannanr@eecs.berkeley.edu

Abstract

We propose a framework for iterative joint source-channel decoding for communication with a fidelity criterion. We consider a class of source models that are used in current state-of-the-art transform image coding schemes. We construct a global graphical model that includes both the channel coding redundancy and the source model and we apply the sum-product algorithm to estimate the transmitted signal with minimum distortion. Our results show the promise of our framework for improving over existing techniques of digital communication.

1. Introduction and Motivation

In modern practical communication systems the source coder is in most cases separate from the channel coder both physically and conceptually. This approach can be inefficient for applications involving the transmission of video, images, speech, etc. This has motivated the research of efficient methods for combining source and channel coders into a joint system. The increased complexity of the problem involving this joint design calls for the formulation of new efficient frameworks that expose most of the advantages of the joint design at a reasonable complexity cost. In this work we propose to tackle the joint source-channel coding problem using a novel *factor graph* framework [5] of iterative decoding.

Factor graphs were initially successfully applied in the area of channel error correction coding, and, specifically, iterative decoding. Turbo decoding and other iterative decoding techniques have in the last few years proven to be landmark developments in coding theory. Factor graphs [5] provide a framework in which iterative algorithms and iteratively decodable codes are easily described. The notion of factor graphs, defined in section 2.1, is related to other graphical models, such as Bayesian networks and Markov Random Fields, and allows an easy integration of the decoding task and other related estimation problems in communication theory such as channel estimation and equalization. The goal of this paper is to extend the area of application of iterative decoding using factor graphs to include the joint source-channel coding. This is accomplished by exploiting the redundancy deliberately left in the source after source encoding together with the controlled redundancy introduced by channel coder in a joint decoding scheme using graphical models. Our graphical source model is of the class of models used in the state-of-the-art image coding algorithms which makes our framework promising for robust image communication.

The idea to utilize the redundancy left in the source after source coding for combating channel noise goes back to Shannon [10]. In [1], the question of how much redundancy should be assigned to the source and channel coders was posed for a binary source with a finite memory transmitted losslessly over a noisy channel. It was shown that for some cases it is beneficial not to compress the source at all in order to avoid the error propagation effects associated with compressing using variable-length entropy coding. In [3] a turbo decoding scheme that uses the channel memory was proposed. This scheme involves the construction of a "supertrellis" on the global state-space of the error correcting code and the channel. A similar approach, in principle, can be applied to the problem considered here by combining states of the code with the states of the source model. The total number of states in this case becomes the product of the numbers of states in the source and in the channel models and can easily get too large to handle. Recently, in [8] the minimum mean-squared error (MMSE) estimation of DPCM-coded images was proposed which modeled the redundancy after the source coding using a Markov mesh model. While being similar to our work, [8] uses a different source model and does not use iterative decoding based on the sum-product algorithm as we suggest in this paper.

2. Problem outline and proposed approach

We consider the problem of communicating a source with a fidelity criterion over a noisy channel with power/bandwidth constraints. In a joint source-channel coding framework, the source and channel encoding and

*This work was supported in part by NSF under grant MIP 97-03181 (CAREER) and by NASA under grant NAG3-2263

0-7803-5700-0/99/$10.00©1999 IEEE

decoding are allowed to be designed jointly to optimize the overall transmission quality subject to the given constraints. The problem that we address in this paper is how to design the joint source-channel decoding of a Hidden Markov sources over an AWGN channel. In this work we use the source symbol error rate (SSER) and the mean-squared error (MSE) to quantify the distortion.

2.1. Factor graphs

Let $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ be a vector of variables. A *factor graph* visualizes the factorization of a global function $f(\mathbf{x})$. Let $f(\mathbf{x})$ factor[1] as $f(\mathbf{x}) = \prod_{i=1}^{m} f_i(\mathbf{x}^{(i)})$, where $\mathbf{x}^{(i)}$ is the set of variables of the function f_i. A factor graph for f is defined as the bipartite graph with two vertex classes V_f and V_v of sizes m and n respectively such that the ith node in V_f is connected to the jth node in V_v iff f_i is a function of x_j. The variables may be from different alphabets as long as the functions f_i are properly defined.

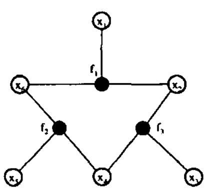

Figure 1. A factor graph for a $[6,3,3]$ binary code C. Function nodes are drawn as small filled circles and variable nodes are drawn as larger non-filled circles.

An example of a factor graph for a $[6,3,3]$ binary code C is shown in Fig. 1. The global function $f(\mathbf{x}) = \prod_{i=1}^{3} f_i(\mathbf{x}^{(i)}) = 1$ if $\mathbf{x}$ is a word in C and $f(\mathbf{x}) = 0$ otherwise. The local functions $f_i(\mathbf{x}^{(i)})$ are equal to 1 if the length three vectors $\mathbf{x}^{(i)}$ have even weight and equal 0 otherwise. For example, $f_1(\mathbf{x}^{(1)}) \equiv f_1(x_1, x_2, x_6) = (x_1 + x_2 + x_6) \bmod 2$. Suppose that a single, randomly chosen codeword of the code C is transmitted over a memoryless channel resulting in the observed vector $\mathbf{y} = (y_1, y_2, \ldots, y_6)$. Then the *a posteriori* distribution of $\mathbf{x}$ is proportional to

$$f(\mathbf{x}) = \prod_{i=1}^{3} f_i(\mathbf{x}^{(i)}) \prod_{j=1}^{6} f_{jj}(y_j|x_j) \qquad (1)$$

where $f_{jj}(y_j|x_j)$ is the channel likelihood function evaluated at the observed value y_j. The factor graph corresponding to the *a posteriori* distribution of $\mathbf{x}$ (not shown) is easily obtained from the original factor graph by connecting each variable to a new observation likelihood function f_{jj} evaluated at the observed values of y_j. The decoding task in the previous example is accomplished by maximizing the cost functions of the form (1). The algorithm which allows us to do it efficiently, though only approximately, is called **the sum-product algorithm**.

2.2. The sum-product algorithm in a tree

In this section we provide an example of the sum-product algorithm applied to a finite tree. A factor graph is a *tree* if and only if there exists a unique path between any two nodes of this graph[2]. In this plot functions are filled circles, observed variables are non-filled circles and unobserved variables are double circles. The factor graph in Fig. 2 is an example of Hidden Markov Model (HMM) with states s_1, s_2, s_3 and observations x_1, x_2, x_3. Function f_1 is the prior distribution of the initial state, functions f_3 and f_5 are conditional distributions of the next state given the present state, functions f_2, f_4 and f_6 are conditional distributions of observations given the state. It is still useful to think about the functions as "checks" though in this case these checks are "soft" rather than "hard" set indicator functions as in our previous example. Suppose we are interested in the *a posteriori* distribution of s_1 after observing the values x_1', x_2', x_3' of x_1, x_2, x_3. The joint density function and its factorization are:

$$f(x_1, x_2, x_3, s_1, s_2, s_3) =$$
$$f_1(s_1) f_2(s_1, x_1) f_3(s_1, s_2) f_4(s_2, x_2) f_5(s_2, s_3) f_6(s_3, x_3)$$

What we want to find is the marginal distribution for s_1 given the observed values x_1', x_2', x_3'. This may be performed efficiently in a distributed manner using the following sum factorization:

$$\begin{aligned} &P(s_1, x_1 = x_1', x_2 = x_2', x_3 = x_3') = \\ &= f_1(s_1) f_2(s_1, x_1 = x_1') \{\textstyle\sum_{s_2} f_3(s_1, s_2) f_4(s_2, x_2 = x_2') \\ &\quad \{\textstyle\sum_{s_3} f_5(s_2, s_3) f_6(s_3, x_3 = x_3')\}\} \end{aligned} \qquad (2)$$

This factorization is what is used in the well-known forward-backward algorithm for HMM's [9] and is an instance of the **sum-product** algorithm. In equation (2) two types of computations are performed: multiplication of local functions and marginalization with respect to local variables. First, f_5 and f_6 are multiplied producing the result which depends on s_2 and s_3. Then this result is summed over s_3 to produce the function of s_2 only. The process repeats for other variables. We can think of the operations just described in terms of *passing messages* from all nodes of the graph along the edges according to some *schedule*. The messages are formed using the following simple rules: a message from a function node to a variable node is the

[1] The definition of multiplication and addition may be taken rather generally. For the precise algebraic properties that multiplication and addition have to satisfy, see [5].

[2] Note, that the factor graph in Fig. 1 have cycles and therefore is not a tree.

product of all messages incoming to the function node with the function itself, marginalized for the variable associated with the variable node; a message from a variable node to a function node is simply the product of all messages incoming to the variable node from other functions connected to it. If the value of the variable is observed, e.g. $x_1 = x_I$ its message is $\delta(x_I)$. It can be shown ([12]) that there exists an optimal (in the sense of the number of messages which are sent) message passing schedule for a tree which gives the *exact* result in a finite number of steps. In coding theory applications the factor graphs are not trees and no optimality of sum-product algorithm is guaranteed. Nevertheless, the performance of the sum-product algorithm in decoding of turbo-codes and low density parity check codes (LDPC [2]) is at present unsurpassed.

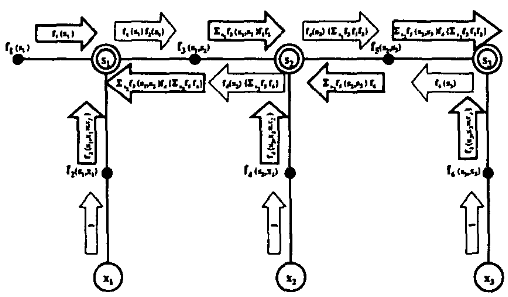

Figure 2. Sum-product algorithm applied to HMM. Thick arrows are messages passed to variables from functions and thin arrows are messages passed to functions from variables.

2.3. Iterative decoding for JSCC

The main contribution of this paper is the novel application of iterative decoding on factor graphs to joint source-channel coding. Suppose that both source and channel bits (or symbols) can be represented in a factor graph form. For example,. channel code parity bits may be generated by a turbo-code or LDPC codes, while the source symbols themselves may be observations of a HMM. Then the joint representation of the source and channel bits/symbols is again a factor graph. Suppose we are interested, as usually, in finding the *a posteriori* distribution on a global graph. In cases of practical importance there is no hope to perform the exact calculation of the joint distribution due to extremely high dimensionality. Instead we propose to use the sum-product message passing algorithm. This idea is illustrated in Fig. 3 where the messages are passed between source and channel models ("checks") in the sum-product algorithm. The higher the memory of the source, the stronger is the influence of the source model on the channel decoding. Unfortunately, as in the case of turbo codes or LDPC codes, one has neither the notion of an optimal message-passing schedule, nor any claims of optimality of the scheme. Our results, however, reveal the benefits of our proposed scheme over conventional approach for certain useful regimes.

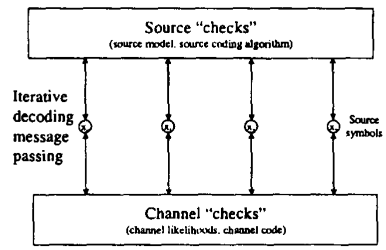

Figure 3. General setup for JSCC with factor graphs. Both source and channel "checks" participate in the message-passing procedure of the sum-product algorithm.

3. JSCC for HMM source using factor graphs

To illustrate the proposed application of the factor graph framework to joint source-channel coding, we assume a discrete-time HMM with Gaussian observations. Consider the source model illustrated in Fig. 4. The underlying Markov chain determines the state of the model. Given the state, the observations are i.i.d. Gaussian random variables with mean and variance indexed by the state. This model is motivated by image transform coding, where similar 2-D models have been proposed for transform coefficients (e.g., schemes in [4, 6] have the same spirit, though not being exactly equivalent to our model here) that acknowledge the fact of remaining localized dependencies even after decorrelating transform.

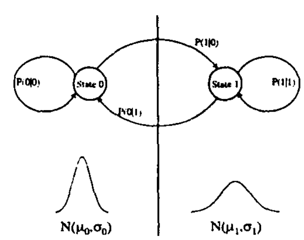

Figure 4. Hidden Markov source model with Gaussian observations.

In this work we assume that the source samples are scalar quantized to produce the source symbols from a finite alphabet[3]. In simulations, we have used a 4-level uniform scalar

[3]This is exactly what is done in the state-of-the-art wavelet image coders.

quantizer. Note, that fixing the quantization in our source is the same as converting it to a finite-alphabet HMM. While we could just start with discrete HMM's from the outset, we emphasize the quantization operation to motivate where this source model can come from.In this work we only consider a one-dimensional source model for simplicity (e.g., when each row of wavelet coefficients is modeled as an HMM), however, the factor graph framework can easily incorporate full two-dimensional model. We consider this for our future research.

In our experiments we use the LDPC channel codes which are currently among the state-of-the-art. LDPC codes were introduced in [2] and have been rediscovered recently in connection with the sum-product algorithm used for their decoding[11, 7]. Suppose the source is to be transmitted over an AWGN channel using LDPC and BPSK with a constraint on the total rate and power, as appropriate, for example, for wireless communication. Since in general the symbols are not independent, it is possible to compress the data stream using a data compression code. Alternatively, the redundancy of the source may be used in lieu of channel redundancy in iterations of the sum-product algorithm. The two options outlined are compared in our experiments.

The reference system uses optimal entropy coding using an arithmetic coder followed by a $(200, 52)$ LDPC channel code[4]. This corresponds to transmitting 52 source encoded bits and 148 parity bits. The channel decoding uses the sum-product algorithm [2] followed by an arithmetic decoder. The proposed joint source-channel coding system deliberately uses fixed length source coding (but avoids the effect of error propagation) and have more source bits to send due to the source memory. We account for this by reducing the amount of channel code parity bits so that the frame size is the same as in the reference system. The decoding in the proposed system is done iteratively using the sum-product algorithm on the global factor graph as shown in Fig. 5. Messages incoming to each source bit are generated both by the channel code, which uses the parity check bits, and by the source model, which exploits the memory in the source. The iterative decoding is stopped either if a valid codeword is detected or if a maximum number of iterations is exceeded. Note, that the sum-product algorithm outputs soft values for each decoded bit, thus effectively approximation the *a posteriori* distribution of the source symbols given the received signals. This, combined with the knowledge of the quantization scheme, allows us to perform the approximate MSE estimation of the transmitted sequence by reconstructing it by the expected value with respect to the calculated distribution.

[4]We have used a short blocklength code in the interests of simulation time complexity.

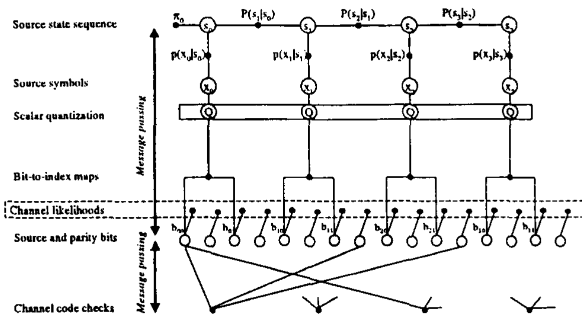

Figure 5. Factor graph for decoding HMM Gaussian source with LDPC code.

3.1. Experimental results

We have simulated the proposed and reference systems and the results are shown in Fig. 6, Fig. 7 and Fig. 8. In Fig. 6 we compare the pure channel code performance - frame error rate (FER) of the three systems. The conventional system uses arithmetic coding and a LDPC code. The reference system does not use entropy coding and has a higher rate channel code for fair comparison. We also show the performance of the proposed system when no source information is used in the sum-product algorithm. Straight lines are the least-squares fits of the corresponding data points (we show data points only for our system) each obtained using one source sequence realization averaged over 100 decoding attempts with different noise realizations. The horizontal axis corresponds to the redundancy of the uncoded source sequence (e.g.,0 - no redundancy, 0.1 - ten percent redundancy). It is clear that the FER performance is the best for the conventional system which is not surprising since the channel factor graph is chosen by design, while the source factor graph is given. In other words we do not expect to improve over LDPC (which already are close to the Shannon's bound) *as channel codes* by leaving some of the redundancy in the source. Note, however, that in the case of a redundant source (upper two curves) using the source memory does significantly improve the channel code performance compared to the case when the memory is present but not used.

In Fig. 7 we compare the source symbol error rate (SSER) between transmitted and received signals. It is interesting to note that at low redundancy it is beneficial not to remove it by entropy coding but instead reduce the rate of the channel code. The reason for this behavior is that our proposed system avoids the well-known error propagation effect present in entropy coded systems. As the redundancy increases, the proposed scheme becomes inferior to the conventional approach. We also plot the simulation results for the approximate MMSE decoding of an HMM

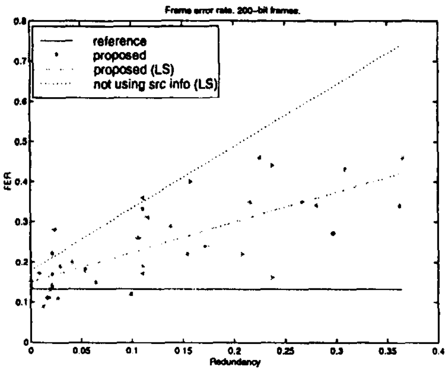

Figure 6. Frame error rate performance of the proposed and reference system. Straight lines are least square fits to the actual data.

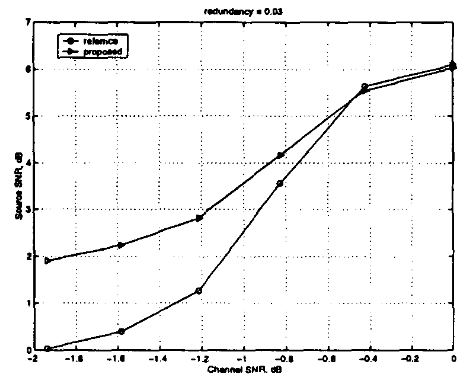

Figure 8. Average source SNR for different AWGN channel noise power. Redundancy in the source is equal to 3%.

approximation of a row data in wavelet image subband of Lena. Parameters of the chain were estimated using the EM-algorithm [9]. In the case of channel errors, decoding was done by reconstructing the samples with their approximate expected values. Notice that the proposed system performance degrades more gracefully, compared to the performance of the fully compressed system and achieves gains of up to 2 dB in the low channel SNR regime.

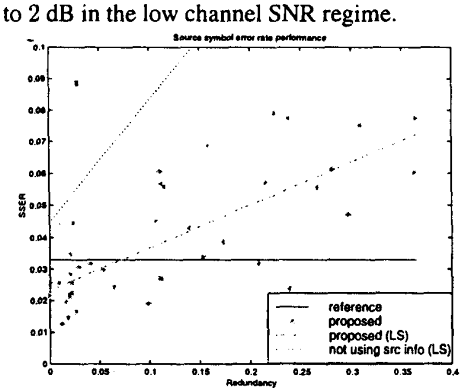

Figure 7. Source symbol error rate performance of the proposed and reference system. Straight lines are least square fits to the actual data.

4. Discussion and future work

In this work we proposed a factor graph framework for joint source-channel coding. The idea is to use the left-over redundancy after the source coding in the joint source-channel decoding using a global graphical model. The sum-product algorithm is applied to this model to produce an estimate of the transmitted signal given both the channel output and the source model. While our source coding model is motivated by real-life applications like typical wavelet image coefficients statistics, in our future work, we plan to broaden the variety of source models and channel codes which fit the proposed factor graph framework. We also consider the two-dimensional image models and incorporate the effects of channels with memory.

References

[1] G. Buch, F. Burket, J. Hagenauer, and B. Kukla. To compress or not to compress? In *Proc. IEEE Glob. Telecommun. Conf.*, 1996.

[2] R. Gallager. Low-density parity-check codes. M.I.T. Press, 1963.

[3] J. Garcia-Frias and J. Villasenor. Exploiting binary markov channels with unknown parameters in turbo decoding. In *Proc. IEEE Glob. Telecommun. Conf.*, pages 3244–3249, Sydney, Australia, November 1998.

[4] R. L. Joshi, T. R. Fischer, and R. H. Bamberger. Optimum classification in subband coding of images. In *Proc. IEEE Int. Conf. Image Proc.*, pages 883–887, Austin, TX, Nov. 1994.

[5] F. Kschischang, B. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *submitted to* IEEE Trans. Inform. Theory, July 1998.

[6] S. LoPresto, K. Ramchandran, and M. T. Orchard. Image coding based on mixture modeling of wavelet coefficients and a fast estimation–quantization framework. *Proc. IEEE Data Compression Conf.*, 1997.

[7] D. MacKay. Good error-correcting codes based on very sparse matrices. *submitted to* IEEE Trans. Inform. Theory, 1997.

[8] M. Park and D. Miller. Improved image decoding over noisy channels using minimum mean-squared estimation and a Markov mesh. *IEEE Trans. Image Process.*, 8:863–867, 1999.

[9] L. Rabiner and B. Juang. *Fundamentals of speech recognition*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[10] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423, 1948.

[11] N. Wiberg. *Codes and decoding on general graphs*. PhD thesis, University of Linköping, Sweden, 1996.

[12] N. Wiberg, H.-A. Loeliger, and R. Koetter. Codes and iterative decoding on general graphs. *Euro. Trans. Telecommun.*, 6:513–526, September 1995.