Paper Submission

Data Compression Conference 2003
Snowbird, Utah, March 25–27

Prof. Martin Cohn
Computer Science Department
Brandeis University
415 South Street
Waltham, MA 02454
USA

Paper Title

"MAP Criterion Trellis Source Coding
for Short Data Sequences"

Authors

Tomas Eriksson, Mirek Novak, John B. Anderson

Index Terms

Lossy source coding, MAP criterion, Trellis coding

Contact Information

Tomas Eriksson
Dept. of Information Technology
Lund University
Box 118
SE–221 00 Lund
Sweden

tomas.eriksson@it.lth.se

Tel. +46 46 222 9763
Fax. +46 46 222 4714

MAP Criterion Trellis Source Coding for Short Data Sequences

Tomas Eriksson, Mirek Novak and John B. Anderson
Department of Information Technology, Lund University
Box 118, SE–221 00, Lund, Sweden
Email: {tomas,mirek,anderson}@it.lth.se

Abstract—**New trellis constructions for compression of short data sequences are presented. Traditional low rate trellis source coding suffer from having a too small a reproduction alphabet. Governed by alphabet-constrained rate-distortion theory we propose a simple method which for a source coding rate of 1 bit per sample allows $2n$ reconstruction values for a trellis with $2n$ states. The proposed method is based on the maximum a posteriori probability criterion distortion measure and it employs tailbiting. Simulation results are compared to TCQ-like methods.**

Index Terms—**Lossy source coding, MAP criterion, Trellis coding.**

1 Introduction

Trellis coding has proven to be an efficient method for data compression. In this work we propose new trellis constructions and use the maximum a posteriori probability (MAP) as a distortion criterion [1]. Simulations show that the proposed method performs as good as, or in some cases even better than, previously reported methods, e.g., [2, 3]. The work is focused on short data sequences and it is shown that tailbiting [5] is a crucial issue in this case.

Trellis coded quantization (TCQ) [2] is a form of trellis coding that labels the trellis branches with subsets of reproduction values. TCQ was inspired by the duality between lossy source coding and modulation methods for digital communications. In particular, TCQ borrow the trellis coded modulation ideas of set partitioning and trellis structure based on convolutional coding. Recent research on TCQ has focused on applications, such as image compression, and not on how to improve the method itself.

TCQ and most other trellis source coding schemes [2, 3, 4] use the Viterbi algorithm (VA) to find the minimum mean squared error (MMSE) path through the trellis. In our work the MAP path is found and hence the BCJR algorithm is used [6]. Also, tailbiting is used for efficient encoding of short data sequences [5]. TCQ uses subsets and a computationally heavy process to optimize the individual reconstruction levels. Here, no subsets are used and all the reconstruction levels are changed merely by multiplying by one scale factor. For a trellis with $2n$ states the current method proposes $2n$ reconstruction levels for rate 1 bit source coding. For the same rate TCQ uses only 4 levels, independently of the number of trellis states. Thus we show methods to increase the reproduction alphabet without changing the source coding rate. This procedure is justified by alphabet-constrained rate-distortion theory [7].

Previous work on MAP source coding has, e.g., been reported in [8] where the MAP criterion was used in the context of residual image coding. In [8] a simple trellis structure based directly on convolutional codes was used in order to construct a joint source–channel coding scheme. Results showed that the method preserves some important visual features better than an MSE based algorithm.

The rest of the paper is organized as follows. In Section 2 we propose new trellis constructions. We also discuss the advantages in MAP source coding, in particular for short data sequences. Simulation results for the uniform and Gaussian sources are given in Section 3 where we also compare the results to TCQ-like methods. Finally our conclusions are drawn in Section 4.

2 Trellis Source Coding

2.1 New Trellis Constructions

Traditionally in trellis source coding the labels and their associated reconstruction levels have been selected by a convolutional encoder structure, see e.g. [2]. However, a good convolutional channel code may not be a good source code. This was shown in e.g. [3]. Convolutional codes and their trellis labels and paths show good pseudo-random properties [9] and this is an asset in source coding. But for source coding, the total number of labels is far too small. For example, a rate 1/2 convolutional channel code used as a rate 1 source code has a total of only 2^2 different labels. According to alphabet-constrained rate-distortion theory, adding more labels alone should be sufficient to improve the performance [7]. In this work we propose a method that allows more labels for the same source coding rate and take from convolutional coding not the labels but only the shift register (SR) transitions.

A memory m SR with 2 outgoing branches from each state has a total of $2 \cdot 2^m$ branches. To combat the problem of having too few reconstruction levels two identical sets of labels, each of size $2n = 2^m$, are used to populate the branches. One set, $\mathcal{L}^u = \{l_k^u\}_{k=0}^{2n-1}$ is used for the up branch and the other, $\mathcal{L}^d = \{l_k^d\}_{k=0}^{2n-1}$, for the down branch. Each label is associated with one reconstruction value. Let the source have zero mean and unit variance. Further, let the distribution of the source be symmetric.

Then if the cumulative distribution function of the source is $F(x)$ the reconstruction levels $\{y_k\}_{k=0}^{2n-1} \in \mathcal{Y}$ are chosen straight-forwardly as

$$y_k \triangleq F^{-1}((2k+1)/(2 \cdot 2n)) \ . \tag{1}$$

In this work we consider only trellises with two outgoing branches from each state. Let $\mathcal{P} = \{p_i\}_{i=1}^{n} = \{(p_i^u, p_i^d)\}_{i=1}^{n}$ denote the set of the two reconstruction values associated with each branch pair. Note that one particular reconstruction value is either in $\mathcal{P}^u = \{p_i^u\}_{i=1}^{n}$ or in $\mathcal{P}^d = \{p_i^d\}_{i=1}^{n}$ and thus $|\mathcal{P}^u \cup \mathcal{P}^d| = 2n$. Now we define the expected value of the cross correlation of the branch pairs as

$$C_{2n} \triangleq \mathbb{E}\{(p_i^u \cdot p_i^d)\} \ . \tag{2}$$

Next we address the question of how to populate the trellis branches with labels, i.e., how to choose and order the sets $\mathcal{P}^u$ and $\mathcal{P}^d$. In total, $\mathcal{P}$ can be ordered in $\frac{(2n)!}{n! \cdot 2^n}$ ways. This expression grows very fast with n and shows that other methods than exhaustive search must be deployed. Thus we need a set of rules that holds regardless of the size of the reconstruction alphabet. Knowing that the source is IID we can formulate

Proposition 1: No labels or transitions should be favored, i.e., the distance between the reconstruction values in the same branch pair should be equal for all branch pairs.

Proposition 2: Each branch pair should span as much of the sample space as possible, i.e., the reconstruction values of each branch pair should be maximally separated.

Propositions 1 and 2 are fulfilled by maximizing the minimum distance between the two reconstruction values associated with each branch pair in $\mathcal{P}$. This requirement is fulfilled if and only if all reconstruction pairs are chosen as $\{(y_k, y_{k+n})\}_{k=0}^{n-1}$, see Figure 1. The proof is not difficult—the problem can, e.g., be recast as maximizing a geometric sum.

For a uniform source on $[-\delta, \delta]$ populated with labels according to the strategy described above, i.e., equally and maximally spread labels, Equation 2 is

$$\begin{aligned} C_{2n} &= \frac{1}{n} \sum_{k=0}^{n-1} y_k \cdot y_{k+n} \\ &= \frac{1}{n} \sum_{k=0}^{n-1} \left(k + \frac{1}{2} - n\right) \frac{\delta}{n} \cdot \left(k + n + \frac{1}{2} - n\right) \frac{\delta}{n} \\ &= -\frac{\delta^2}{6} - \frac{1}{12} \frac{\delta^2}{n^2} \ . \end{aligned}$$

For the unit variance uniform source $\delta = \sqrt{3}$ and hence

$$\lim_{n \to \infty} C_{2n} = -\frac{1}{2} \ .$$

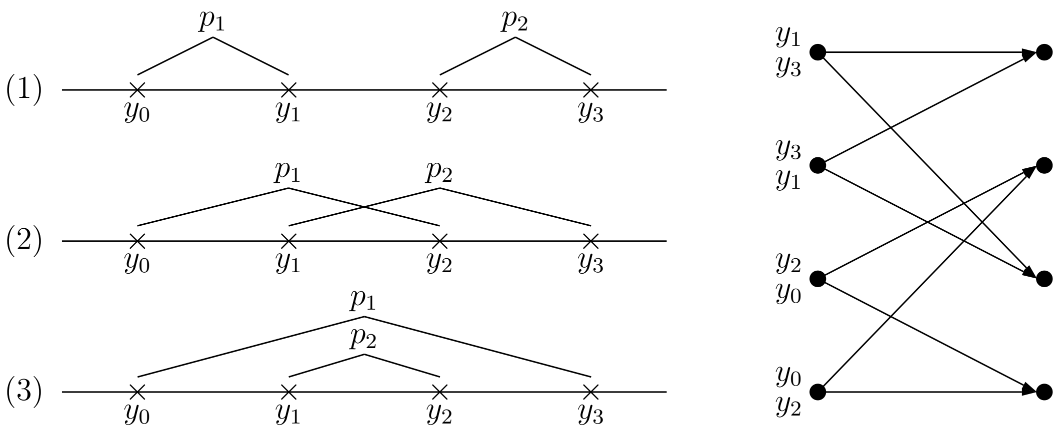

Figure 1: Left: Illustration of all label pair combinations for four different labels. Right: Trellis segment for the maximum minimum distance constellation (construction (2) in the left figure).

We have shown that, with the strategy described above for selecting label pairs, the expected value of the cross correlation of the branch pair reconstruction values approaches $-1/2$ for large reconstruction alphabets. For a non-uniform source we do not have precisely equal distances between reconstruction values, but similar arguments lead approximately to the same conclusion.

The algorithm used to find the minimum distortion path uses both forward and backward recursions to obtain soft information values regarding the source sequence—see Section 2.2. Hence we propose a mirror property on the trellis label structure. This is justified by the fact that the source is stationary and therefore has the same statistical properties regardless of the direction of time. Consider each four-tuple of connected branches in a trellis segment and denote by $g : \mathcal{L} \to \mathcal{Y}$ a one-to-one mapping from the space of trellis labels to the space of real numbers. If the two branches exiting state i and entering states i' and j' are labeled $l_i^u = g^{-1}(y_{k_1})$ and $l_i^d = g^{-1}(y_{k_2})$, respectively, the two branches exiting state j and entering states i' and j' must satisfy

Proposition 3:

$$\begin{aligned} l_j^u &= l_i^d \\ l_j^d &= l_i^u \ , \end{aligned}$$

$l_i^u, l_j^u \in \mathcal{L}^u$ and $l_i^d, l_j^d \in \mathcal{L}^d$.

We call this a butterfly structure—see Figure 2 for the general case and the right part of Figure 1 for a four state realization. Using the butterfly structure we obtain the desired omni-directional trellis.

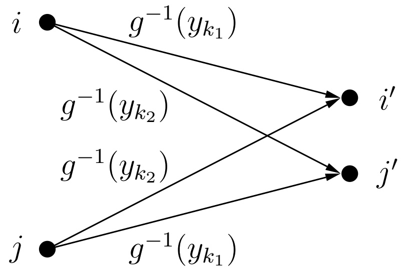

Figure 2: Part of a trellis segment with labels arranged according to the butterfly structure.

2.2 MAP Trellis Coding

In general the problem of trellis source coding consists of finding the minimum distortion path for a given input and a given trellis structure. It is well known that for the MMSE criterion, the Viterbi algorithm (VA) delivers the MMSE path. In this work we focus on the compression of short data sequences. Typical communication systems utilizing small packets of data are packet switched networks and mobile communications.

If the VA starts in the all-zero state, a start-up phase is necessary for the trellis to be fully developed; the starting state can also be specified explicitly at the cost of m bits. If the VA ends in the all-zero state the encoder must be driven into this state, a procedure which also degrades the performance. In theory the lengths of the start-up and termination phases are both equal to the memory length of the code, i.e., the number of SR delays ($= m$). However, simulations have shown that for source coding these lengths are much larger in practice. For example, for a memory 3 code both the start-up and termination phases can in some cases be as long as 15–20 symbols.

Hence for short data sequences it is crucial to find methods that do not suffer from rate loss and/or do not degrade the performance at the beginning or end of each frame. The principle of tailbiting [5] prevents all rate loss and most of the performance loss by allowing the start and stop states to be chosen arbitrarily, as long as they are the same. However, there are a number of reasons not to use the VA or similar dynamic programming methods in the context of tailbiting. First, the complexity of a true tailbiting VA is 2^{2m}. Second, approximate VAs with complexity equal to some constant multiple of 2^m have been devised, but these can have convergence problems—see, e.g., [10].

We now use the maximum a posteriori probability (MAP) as a distortion measure. An algorithm that finds the symbol-by-symbol MAP path for trellis codes, with complexity $2 \cdot 2^m$, is the BCJR algorithm [6]. The BCJR algorithm can easily be adopted to tailbiting (T–BCJR) with complexity approximately $2 \cdot 2^m$ as shown in [1]. The T–BCJR approach has rapid, guaranteed convergence [11] and only two times the computation of the non-tailbiting VA. Another advantage is that MAP allows single outliers in the reconstruction as shown in [8]. The MSE metric does not allow this, and instead tries to find a compromise resulting in the smallest average distortion. When using a MAP encoder, shot-noise can appear. However, shot-like noise can easily be removed by a median filter.

Consider the source encoding of a sequence $\boldsymbol{x}= x_1 x_2 \ldots x_N$. All x_t represent one (real) symbol. Any trellis encoder will produce a sequence of labels, $\boldsymbol{\Lambda}= \Lambda_1\Lambda_2 \ldots \Lambda_N$, where each label is a function of the present and the next trellis state, i.e., $\Lambda_t = f(S_t, S_{t+1})$. Denote by $\boldsymbol{L}$ a realization of the sequence $\boldsymbol{\Lambda}$. Then the quantization error $\boldsymbol{q}$ is

$$\boldsymbol{q} = \boldsymbol{x} - g(\boldsymbol{L}) \ , \tag{3}$$

where g is defined according to Section 2.1. With the MAP criterion, finding the optimum of $\boldsymbol{q}$ in Equation 3 is equivalent to the following: Given $\boldsymbol{x}$ find an outcome $\boldsymbol{L}$ of $\boldsymbol{\Lambda}$ such that

$$\Pr\{\boldsymbol{L} \mid \boldsymbol{x}\} \cdot \Pr\{\boldsymbol{x}\} \tag{4}$$

is maximized.

The BCJR algorithm outputs the probability that the state i is reached at time t, given the observed input sequence $\boldsymbol{x}$. If S_t is the state at time t, we can write this probability as

$$\Pr\{S_t = i \mid \boldsymbol{x}\} \ . \tag{5}$$

The algorithm does not compute this probability directly, but instead computes

$$\begin{aligned} \lambda_t(i) &\stackrel{\triangle}{=} \Pr\{S_t = i, \boldsymbol{x}\} \\ &= \Pr\{S_t = i \mid \boldsymbol{x}\} \cdot \Pr\{\boldsymbol{x}\} \ . \end{aligned} \tag{6}$$

Since the values of $\boldsymbol{L}$ are defined by a description of the path from state i at time $t-1$ to state j at time t in the trellis, we have a well defined mapping, $\{\boldsymbol{L}\} \to \{S_t = i\}$, between labels and states, i.e., Equations 4 and 6 are equivalent and hence, the BCJR algorithm can be utilized for a source coding purpose.

The probability of a state transition from state i at time $t-1$ to state j at time t in the trellis can be written as

$$\begin{aligned} S_t(i \to j) &\stackrel{\triangle}{=} \Pr\{S_{t-1} = i, S_t = j, \boldsymbol{x}\} \\ &= \alpha_{t-1}(i)\Gamma_t(i,j)\beta_t(j) \ , \end{aligned}$$

where $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\Gamma}$ have been introduced as auxiliary variables. With the aid of these variables the desired $\boldsymbol{\lambda}$ values can be found using both forward and backward recursions. From above

$$\begin{aligned} \Gamma_t(i,j) &\stackrel{\triangle}{=} \Pr\{S_t = j, x_t \mid S_{t-1} = i\} \\ &= \Pr\{x_t \mid i \to j\} \Pr\{i \to j\} \quad t = 1, \ldots, N \\ \alpha_t(i) &\stackrel{\triangle}{=} \Pr\{S_t = i, x_1 \ldots x_t\} \qquad t = 1, \ldots, N \\ \beta_t(j) &\stackrel{\triangle}{=} \Pr\{x_{t+1} \ldots x_N \mid S_t = j\} \quad t = 1, \ldots, N-1 \ . \end{aligned}$$

The equality in the expression for $\mathbf{\Gamma}_t$ follows from an assumption that the trellis evolves in a Markov fashion and that the introduced quantization noise is memoryless. The $\mathbf{\Gamma}_t$ matrices serve as input to the algorithm. For an input with Gaussian distribution we assume that $\mathbf{\Gamma}_t$ has also a Gaussian form

$$\Gamma_t(i,j) = \exp\left(-\frac{(x_t - g(L_t))^2}{c_1}\right) , \tag{7}$$

where c_1 is free parameter, $c_1 \approx .1$. For an illegal path in the trellis, $\Gamma_t(i,j) = 0$. The row vectors $\boldsymbol{\alpha}_t$ and column vectors $\boldsymbol{\beta}_t$ are computed recursively as

$$\begin{aligned} \boldsymbol{\alpha}_t &= \boldsymbol{\alpha}_{t-1}\mathbf{\Gamma}_t, \quad && t = 1, \ldots, N \\ \boldsymbol{\beta}_t &= \mathbf{\Gamma}_{t+1}\boldsymbol{\beta}_{t+1}, && t = N-1, \ldots, 1 . \end{aligned}$$

For tailbiting these recursions are repeated until convergence. Then the elements of the row vectors $\boldsymbol{\lambda}_t$ are

$$\lambda_t(i) = \alpha_t(i) \cdot \beta_t(i), \quad t = 1, \ldots, N .$$

From the $\boldsymbol{\lambda}$ sequence, the path in the trellis is determined as the transitions between the states with highest probability at each time instant, given that these are legal transitions in the trellis. If Equation 7 is close to the true $\mathbf{\Gamma}_t$, then $\boldsymbol{\lambda}$ describes the approximate MAP path as the representation of the source sequence $\boldsymbol{x}$.

3 Results

The new trellis constructions were simulated with uniform and Gaussian sources with unit variance. In both cases the initial set of reconstruction values was chosen according to Equation 1. This set was then multiplied by scalars in a set $\mathcal{Z}$, and tested with an input training sequence. Then z_{opt} was chosen as the $z \in \mathcal{Z}$ yielding the optimum performance. For all trellis sizes tests show that for the uniform source $z_{\text{opt}} \approx .88$ and for the Gaussian source $z_{\text{opt}} \approx .98$. Finally, the trellises were tested with the scaled set of reconstruction values and another set of input samples. To insure statistical validity the total length of the input segment was in all cases 200 000 samples.

Tables 1 and 2 compare the performance to the results reported in [2] and [3]. Both [2] and [3] use the TCQ approach and computationally heavy optimization procedures to find the individual reproduction levels. Also the VA is used to find the MMSE path. It is seen that for all but the smallest trellises the new method performs better. This is because for the same rate, a larger reproduction alphabet is used. For the smallest trellises the new constructions suffer from not having individually optimized levels. We have also compared the tailbiting BCJR algorithm to the VA for sequences of shorter data blocks. For a fair comparison the trellises, labels and reconstruction values were identical for the VA and the T–BCJR experiments. The results are shown in Tables 3 and 4. It is clear that here the T–BCJR algorithm outperforms the VA for all trellis sizes. All constructions in the tables fulfill Propositions 1–3, as presented in Section 2.1. We have also tested a very large number of trellises of size $2n$ and with $2n$ reconstruction values from Equation 1 but not populated according to these rules, i.e., randomly populated. The results are poor and show the importance of having a structured population.

# states	T–BCJR	[2]	[3]
4	6.04	6.22	6.25
8	6.20	6.33	6.32
16	6.38	6.39	6.37
32	6.45	6.44	6.43
64	6.51	6.48	6.47
128	6.60	6.55	6.52
256	6.70	6.58	6.56

Table 1: SNR values in dB for the uniform source, $D_{\mathrm{U}}(R) = 6.79$ dB. Input block length = 1024 symbols.

# states	T–BCJR	[2]	[3]
4	4.68	5.00	5.05
8	5.11	5.19	5.19
16	5.30	5.27	5.30
32	5.49	5.34	5.39
64	5.72	5.43	5.49
128	5.80	5.52	5.56
256	5.91	5.56	5.63

Table 2: SNR values in dB for the Gaussian source, $D_{\mathrm{G}}(R) = 6.02$ dB. Input block length = 1024 symbols.

# states	4	8	16	32	64	128	256
VA	5.66	5.78	5.97	6.02	6.07	6.13	6.21
T–BCJR	6.03	6.18	6.35	6.38	6.40	6.45	6.52

Table 3: SNR values in dB for the uniform source, $D_{\mathrm{U}}(R) = 6.79$ dB. Input block length = 64 symbols. The VA starts in the all zero state.

# states	4	8	16	32	64	128	256
VA	4.11	4.41	4.67	4.82	5.02	5.13	5.28
T–BCJR	4.56	5.07	5.22	5.34	5.51	5.62	5.71

Table 4: SNR values in dB for the Gaussian source, $D_{\mathrm{G}}(R) = 6.02$ dB. Input block length = 64 symbols. The VA starts in the all zero state.

4 Conclusions

In this work we have presented some simple new methods for trellis source coding. The focus on short data blocks and the MAP criterion led to the use of the tailbiting BCJR algorithm. Simulation results showed the strong superiority of the T–BCJR algorithm over VA in this case. We have also compared the results to TCQ-like methods and showed that by allowing a larger reconstruction alphabet the new constructions can perform better. This is especially true for large trellises.

References

[1] J.B. Anderson and S.M. Hladik, "Tailbiting MAP decoders," *IEEE Journ. Sel. Areas Comm.,* vol. 16, No. 2, pp. 297–302, Feb. 1998.

[2] M.W. Marcellin and T.R. Fischer, "Trellis coded quantization of memoryless and Gauss-Markov sources," *IEEE Trans. Comm.,* vol. 38, No. 1, pp. 82–93, Jan. 1990.

[3] R.J. van der Vleuten and J.H. Weber, "Construction and evaluation of trellis-coded quantizers for memoryless sources," *IEEE Trans. Inform. Theory,* vol. 41, No. 3, pp. 853–859, May, 1995.

[4] R. Lariola and N. Farvardin, "Trellis-based scalar vector quantizer for memoryless sources," *IEEE Trans. Inform. Theory,* vol 40, pp. 860–870, May 1994.

[5] H.H. Ma and J.K. Wolf, "On tail biting convolutional codes," *IEEE Trans. Comm.,* vol. COM-34, No. 2, pp. 104–110, Feb. 1986.

[6] L.R. Bahl *et al.*, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory,* vol. IT-20, pp. 284–287, Mar. 1974.

[7] W.A. Finamore and W.A. Pearlman, "Optimal encoding of discrete time continuous-amplitude memoryless sources with finite output alphabets," *IEEE Trans Inform. Theory,* vol IT-26, No. 2, pp 144–155, Mar., 1980.

[8] T. Eriksson *et al.*, "Image coding with the MAP criterion," in Proc. *Data Compression Conference*, p. 453, Snowbird, Utah, April, 2002.

[9] R. Johannesson and K.Sh. Zigangirov, *Fundamentals of Convolutional Coding,* IEEE Press, New York, NY, 1999.

[10] J.B. Anderson and S.M. Hladik, "An optimal circular viterbi decoder for the bounded distance criterion," *IEEE Trans. Comm.,* vol. 50, No. 11, Nov. 2002.

[11] J.B. Anderson and K. Tepe, "Properties of the tailbiting BCJR decoder," in *Codes, Systems and Graphical Models,* Marcus and Rosenthal (eds.), Springer Verlag, New York, NY, 2001.