New Trellis Source Codes Based on Linear Congruential Recursions

John B. Anderson, Tomas Eriksson, Mirek Novak[1]
Dept. of Information Technology, Lund University
P.O. Box 118, SE–221 00 Lund, Sweden
e-mail: {anderson, tomas, mirek}@it.lth.se

Abstract — **We evaluate new rate–distortion codes based on simple linear congruential recursions. Codes are demonstrated for the Uniform and Gaussian sources and their performance exceeds that so far reported in the literature, including TCQ.**

I. Introduction

In a rate–distortion trellis source code, the branches of the trellis are populated with reproducer letters $\hat{z}$, or in the case of trellis coded quantization (TCQ), with sets of letters. If source outcomes are to be adequately represented, the alphabet of $\hat{z}$ values must be rich enough and the $\hat{z}$ must occur in a suitable distribution. According to classical alphabet–constrained rate–distortion theory, a reproducer alphabet of 8 or more values is required. For the mean square error (MSE) criterion and for Gaussian sources, their frequency of occurrence should be approximately Gaussian. Furthermore, reproducer values down a trellis path should be pseudorandom. We find that codes meeting all of these requirements can be generated by simple linear congruential recursions.

II. Linear Congruential Codes

A classic way to generate rate–distortion codes has been to borrow the convolutional shift register from channel coding, and map the c-bit convolutional trellis labels to 2^c reproducer values. This is both a good and a bad idea. Convolutional trellis paths show strong pseudorandom properties. But it is clumsy to obtain large reproducer alphabets, and codes with rates $\neq 1$ are tedious to construct. A linear congruential (LC) recursion creates pseudorandom codewords easily and with more of the right properties. An LC output sequence satisfies

$$\lambda_{n+1} = (g\lambda_n + a) \bmod M \qquad \lambda_n \in \{0,\ldots,M-1\}, \qquad (1)$$

where M is the number of states in the recursion, and $a, g \in \{0,\ldots,M-1\}$. We will assume that M, g, a are such that (1) progresses through all the integers in $\{0,\ldots,M-1\}$, so that it makes sense to take λ_n as the state of the LC recursion. We can generate trellis branch labels in the following way. Suppose a node in the code trellis has trellis state s. We can set the state of (1), i.e., λ_n, equal to s, iterate the recursion, and take the label as λ_{n+1}. If there are several reproducer values on each branch ($R < 1$), the recursion is repeated. If $b \geq 2$ branches stem from each node ($R \geq 1$), we can associate separate recursions based on $(g_0, a_0), \ldots, (g_{b-1}, a_{b-1})$ with each branch out of the node. Many ways can be imagined to generate labels with an LC recursion, but this one works well.

To make a source code we must first map an alphabet of M reproducer values $\{\rho_0,\ldots,\rho_{M-1}\}$ to the M labels. We simply associate $\rho_0,\ldots,\rho_{M-1}$ in order with $0,\ldots,M-1$. Second, we must select the alphabet. This we do by a standard method in random number generation, dividing the source distribution into M equal pieces: for the source cumulative distribution function $F(z)$, $\rho_i = F^{-1}((2i+1)/M), \quad i = 0,\ldots,M-1$. If the labels are random, this creates a distribution of reproducer values that is independent and that matches the source. The method here adapts to any reasonable source. Finally, we must specify the state transitions in the trellis. This may also be done with an independent LC recursion or by the shift register method inherent in convolutional encoders. We tested both options.

A effective way to find good LC codes is to select them at random, evaluate the MSE of each, and build up a list of best codes. To make random selection more precise, one must control carefully the experimental rms error σ in the MSE and the bias introduced by keeping a best list. Table 1 shows the results at rate 1 bit/sample. It compares the best known previous MSE results, as far as we know, for trellis encoding of these sources, as reported in [1, 2, 3]. These papers use a TCQ approach and also attempt to optimize the precise reproducer values. The outcomes represent 200–10000 length-100 frames of source data, to produce the σ shown. The code trellis is searched by the Viterbi algorithm. MSE is not recorded during the startup of the frame, a strategy that gives the eventual MSE achieved with long frames.

III. Conclusions

(i) For the same state size, we improve most results in [1, 3] except for small codes with the Uniform source. This is because we make no attempt to optimize reproducer values, which degrades codes with a small reproducer set, but hardly affects large codes. (ii) Straightforward LC codes generally perform better than TCQ approaches. (iii) Without exception, a random search for good LC parameters yields that the multipliers g_0 and g_1 should be the same, with different offsets a_0 and a_1. (iv) All good codes have a_0 and a_1 such that the normalized cross correlation of the sets of reproducer values $\rho_0,\ldots,\rho_{M-1}$ from the up and down branches is close to $-1/2$. This finding contradicts the accepted wisdom that trellis source code branches should be antisymmetrical (correlation -1), or independent (correlation 0).

#	Uniform			Gaussian		
States	M–F	V–W	LC	M–F	V–W	LC
4	0.239	0.237	0.247	0.293*	0.313	—
8	0.233	0.233	0.245	0.303	0.303	0.301
16	0.230	0.231	0.239	0.297	0.295	0.286
32	0.227	0.228	0.232	0.273*	0.289	0.276
64	0.225	0.225	0.225	0.286	0.283	0.269
128	0.221	0.223	0.220	0.281	0.278	0.262
256	0.220	0.221	0.215	0.278	0.274	0.254
512	—	—	0.2105	—	0.270	0.2503

Table 1: MSE for best LC codes found compared to best codes in [1, 2, 3]. $\sigma = .001$–$.002$ ($\sigma = .00010$ for 512 states). For Uniform source $D(R) \approx .2094$; Gaussian $D(R) = .2500$. *Result in [2].

References

[1] M.W. Marcellin and T.R. Fischer, "Trellis coded quantization of memoryless and Gauss-Markov sources," *IEEE Trans. Commun.*, vol. COM-38, pp. 82–93, Jan. 1990.

[2] L. Yang and T.R. Fischer, "A new trellis source code for memoryless sources," *IEEE Trans. Information Theory*, vol. IT-44, pp. 3056–3063, Nov. 1998.

[3] R.J. van der Vleuten and J.H. Weber, "Construction and evaluation of trellis-coded quantizers for memoryless sources," *IEEE Trans. Information Theory*, vol. IT-41, pp. 853–859, May 1995.

[1] This work was supported by the Swedish Foundation for Strategic Research — Personal Computing and Communications.