Hexacode-Based Quantization of the Gaussian Source at $1/2$ Bit Per Sample

Stéphane Ragot, Jean-Pierre Adoul, and Roch Lefebvre

***Abstract*—We present the performance of several suboptimal algebraic quantizers in 24 dimensions. The Gaussian source is encoded at $1/2$ bit per sample using the binary extended Golay code $\mathbb{C}_{24}$ and the hexacode H_6. We also propose two new suboptimal decoding algorithms for the hexacode H_6.**

***Index Terms*—Golay code, hexacode, vector quantization.**

I. Introduction

THE objective of this letter is to compare nearest-neighbor and suboptimal algebraic quantizers in terms of performance/complexity tradeoff.

We study the discrete-time memoryless Gaussian source, because it can characterize well several whitened signals such as prediction or quantization residues. It is also a major yardstick in quantization theory and the rate-distortion bound is known explicitly [1]. Since in high dimensions Gaussian source vectors are essentially located on a thin spherical shell, we consider only a spherical codebook. We study the case of the extended binary Golay code $\mathbb{C}_{24}$ mapped on the unit Euclidean sphere by the $\{0,1\} \rightarrow \{-1,1\}$ operation, and by normalizing the resulting points by $1/\sqrt{24}$ [2]. This code describes a source vector in 24 dimensions with 12 b. The nearest-neighbor search in such a codebook can be made fast, because the search becomes equivalent to dot product maximization. The number of additions and comparisons is minimized by exploiting the underlying code structure—typically by a trellis search [2].

The hexacode H_6 is used to carry out the investigation. It has been shown recently that it captures in a few codewords the essence of the Golay code $\mathbb{C}_{24}$ [3], [4] and the Leech lattice [5]. Indeed, these two exceptional structures can be reduced to H_6 using *ad hoc* projections exhibited in [3], [4], and [5]. The hexacode is used herein to cut efficiently the complexity of the nearest-neighbor search while retaining much of the quantization performance of the Golay code.

The paper is organized as follows. Section II describes briefly the decoding of the hexacode H_6 and presents two new suboptimal algorithms. Section III reviews briefly how to decode efficiently the Golay code using H_6 and presents numerical results, before concluding.

II. Hexa-Decoding

The hexacode H_6 is defined as the linear quaternary code (6, 3, 4) specified, for instance, by the generator matrix [3]

$$\begin{bmatrix} 1 & & & 1 & \bar{\omega} & \omega \\ & 1 & & 1 & \omega & \bar{\omega} \\ & & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Its symbols $\{0,1,\omega,\bar{\omega}\}$, with $\bar{\omega}=\omega^2$, are the elements of the Galois field GF(4). The addition and multiplication over GF(4) extend their binary analogs [4], [6] and can be derived from the primitive polynomial $\pi(\omega)=1+\omega+\omega^2$ of GF(4).

In what follows, the term hexa-decoding refers to as the decoding of the hexacode. The hexa-decoding problem can be formulated as follows [4]. Given the confidence values $\mu_i(x_i)$ for the quaternary symbols $x_i \in \{0,1,\omega,\bar{\omega}\}$ and $i \in \{1,2,\ldots,6\}$, find the codeword $\hat{x} \in H_6$ which maximizes the metric $M(x)$. That is, find

$$\hat{x} = \arg\max_{x \in H_6} M(x) \tag{1}$$

where the metric is defined as

$$M(x) = \mu_1(x_1) + \cdots + \mu_6(x_6) \tag{2}$$

and x_i denotes the ith element of x.

Optimal hexa-decoding algorithms have been proposed in [4], [7], and [8]. The winner $\hat{x}$ can be found efficiently using a trellis, and this technique requires as few as 179 operations per source vector (116 additions, 63 comparisons) with a 4-section trellis as described in [7]. This complexity may be still too high for implementation. Therefore, several bounded-distance hexa-decoding algorithms have been proposed in [9] and [10]. They are all based on the same preselection step: a hard decision.

We propose here a new class of suboptimal hexa-decoding algorithms. They rely on the following property of the hexacode: if an arbitrary point $x=(x_1,\ldots,x_6)$, with $x_i \in$ GF(4) for $i \in \{1,\ldots,6\}$, is not in the hexacode H_6, there are 11 hexa-codewords at Hamming distance ≤ 3 from x. The algorithm is detailed according to [11] the following.

1) Compute the codeword $\bar{x}$ whose components are defined as

$$\bar{x}_i = \arg\max_{x_i \in \{0,1,\omega,\bar{\omega}\}} \mu_i(x_i), \quad \text{for } i \in \{1,2,\ldots,6\}. \tag{3}$$

   If the hard-decoded point $\bar{x}$ is a hexa-codeword, then select it as the winner $\hat{x}=\bar{x}$, and stop.

2) Otherwise, identify the 11 hexa-codewords at Hamming distance ≤ 3, that is, sharing at least 3 symbols with $\bar{x}$, calculate their metrics, and select a hexa-codeword of optimal metric.

This algorithm requires 83 operations at most. Complexity can be reduced further down to at most 35 operations by examining the hexa-codewords at Hamming distance ≤ 2 only. These two versions of the algorithm are referred hereafter to as depth-first search 1 and 2, respectively.

Paper approved by S. G. Wilson, the Editor for Coding Theory and Applications of the IEEE Communication Society. Manuscript received July 15, 2000; revised February 12, 2001.

The authors are with the Department of Electrical & Computer Engineering, University of Sherbrooke, Sherbrooke, QC J1K 2R1, Canada (e-mail: ragot@gel.usherb.ca; adoul@gel.usherb.ca; lefebvre@gel.usherb.ca).

Publisher Item Identifier S 0090-6778(01)10609-4.

TABLE I
QUANTIZATION PERFORMANCE OF THE GOLAY CODE

Algorithm	SNR (dB)	Complexity
trellis of cubing construction [8]	2.51	1416
trellis of the hexacode [4]	2.51	651
trellis of the hexacode [7]	2.46	431
Amrani-Be'ery 1 [10]	2.45	351
Amrani-Be'ery 2 [10]	2.43	275
depth-first search 1	2.45	303
depth-first search 2	2.33	207

III. PERFORMANCE AND COMPLEXITY OF GOLAY CODE QUANTIZATION

The relationship between the Golay code and the hexacode is described in [3] and [4] using a binary 4×6 array known as the Miracle Octad Generator (MOG). In short, the Golay code can be defined as the code whose projection from the MOG is the hexacode and which verifies two parity constraints. These constraints are referred to as global parity and top-row parity with respect to the MOG array [4]. This point of view is equivalent to a multilevel construction [7] and yields the generator matrix found at the bottom of the page. Decoding the Golay code then amounts to parsing a trellis of the hexacode once for each global parity and checking for the top-row parity in every path [4]. Very efficient decoding algorithms for the Golay code can be derived by relaxing some constaints in the optimal decoding. For instance, the top-row parity could be checked only after complete parsing of the trellis [7]. Moreover a suboptimal hexa-decoding algorithm could be used instead of trellis decoding [9], [10].

A Monte Carlo simulation was used to estimate the mean-squared error (MSE) for the decoding algorithms of the Golay code described in this paper. The gain-shape reconstruction v of the Gaussian i.i.d process $u \in \mathbb{R}^{24}$ is given at $1/2$ bit per sample by $v = g\ c$ where g is a constant and $c \in \mathbb{C}_{24}$. The optimal performance is bounded by the Shannon limit $\log_{10}(2) \approx 3.01$ dB in terms of signal-to-noise ratio (SNR), and the gain g is optimized *a posteriori* as in [12]. Note that in [12] approximately 10 000 Gaussian vectors were used to evaluate the performance of optimal Golay quantization. We generated a sequence of 100 000 Gaussian vectors with the Box–Muller method [13] for this paper. Table I shows the results for different decoding algorithms of the Golay code. Complexity is measured as the number of floating-point operations (additions and comparisons). The suboptimal algorithms perform favorably.

IV. CONCLUSION

Algebraic codes yield efficient quantization techniques, which require virtually no storage and few floating-point operations compared to unstructured vector quantizers. However, the complexity of the optimal nearest-neighbor search in algebraic codebooks explodes very quickly with the codebook dimension. This paper presented some numerical results that motivate the further development and performance analysis of very high-dimensional quantization codebooks with suboptimal algebraic techniques.

Furthermore, the decoding of the hexacode can be used not only for Golay quantization but also for the Leech lattice decoding problem [5], [7], [9], [10]. It would be interesting to apply the two suboptimal hexa-decoding algorithms described in this paper to the Leech lattice.

ACKNOWLEDGMENT

The authors would like to thank A. Vardy for providing some references at the early stage of this work, as well as O. Amrani and Y. Be'ery for some stimulating correspondence. They are also grateful to V. T. Ruoppila for reviewing the manuscript thoroughly and to C. Lamblin for helpful discussions regarding the numerical results.

REFERENCES

[1] T. Berger, *Rate Distortion Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1971.
[2] Z. Ben-Néticha, P. Mabilleau, and J.-P. Adoul, "The "stretched"-golay and other codes for high SNR finite-delay quantization of the Gaussian source at $1/2$ bit per sample," *IEEE Trans. Commun.*, vol. 38, pp. 2089–2093, Dec. 1990.
[3] V. Pless, "Decoding the golay codes," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 561–567, July 1986.

1111	0000	0000	0000	0000	1111	top-row parity constraint [4]
0000	1111	0000	0000	0000	1111	
0000	0000	1111	0000	0000	1111	
0000	0000	0000	1111	0000	1111	
0000	0000	0000	0000	1111	1111	
0101	0000	0000	0101	0011	0110	hexacode mapped to GF(2)
0011	0000	0000	0011	0110	0101	
0000	0101	0000	0101	0110	0011	
0000	0011	0000	0011	0101	0110	
0000	0000	0101	0101	0101	0101	
0000	0000	0011	0011	0011	0011	
0111	0111	0111	0111	0111	1000	glue

[4] A. Vardy and Y. Be'ery, "More efficient soft decoding of the golay codes," *IEEE Trans. Inform. Theory*, vol. 37, pp. 667–672, May 1991.

[5] ——, "Maximum likelihood decoding of the leech lattice," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1435–1444, July 1993.

[6] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, 3rd ed. Berlin: Springer-Verlag, 1998.

[7] O. Amrani, Y. Be'ery, A. Vardy, F.-W. Sun, and H. C. A. van Tilborg, "The leech lattice and the golay code: Bounded-distance decoding and multilevel constructions," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1030–1043, July 1994.

[8] G. D. Forney, "Coset codes—Part II: Binary lattices and related codes," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1152–1187, Sept. 1988.

[9] A. Vardy, "Even more efficient bounded-distance decoding of the hexacode, the golay code, and the leech lattice," *IEEE Trans. Inform. Theory*, vol. 41, pp. 1495–1499, Sept. 1995.

[10] O. Amrani and Y. Be'ery, "Efficient bounded-distance decoding of the hexacode, the golay code, and the leech lattice," *IEEE Trans. Inform. Theory*, vol. 44, pp. 534–537, May 1996.

[11] S. Ragot, "The Hexacode, the Golay Code and the Leech Lattice: Construction, Decoding, Application to Quantization," M.A.Sc. Thesis (in French), Univ. of Sherbrooke, Depr. of Electrical Engineering, Sherbrook, QC, Canada, Dec. 1999.

[12] J.-P. Adoul and C. Lamblin, "A comparison of some algebraic structures for CELP coding of speech," *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4, pp. 1953–1956, 1987.

[13] W. H. Press, *Numerical Recipes in C*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1992.