Lapped Orthogonal Vector Quantization

Henrique S. Malvar
PictureTel Corporation
Danvers, MA 01923

Gary J. Sullivan
PictureTel Corporation
Danvers, MA 01923

Gregory W. Wornell
Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

The blocking artifacts that arise in the use of traditional vector quantization (VQ) schemes can, in general, be virtually eliminated via an efficient lapped VQ strategy. With lapped VQ, blocks are obtained from the source in an overlapped manner, and reconstructed via superposition of overlapped codevectors. The new scheme, which we term lapped orthogonal vector quantization (LOVQ), requires no increase in bit rate and, in contrast to other proposed approaches, no significant increase in computational complexity or memory requirements. Attractively, the use of LOVQ also leads to a modest increase in coding gain over traditional VQ schemes of comparable complexity.

1 Introduction and Background

Vector quantization (VQ) plays an important role in a wide range of signal coding and data compression applications [1]. In a typical application, involving imagery or speech for example, the signal is partitioned into contiguous blocks of equal size, each of which corresponds to a vector of signal samples. Each vector is then represented by one of a set of candidate codevectors that is closest to the vector with respect to some distortion measure. This set—the codebook—is available to the decoder as well, so for each block only the index of the codevector need be transmitted to allow suitable reconstruction of the block at the receiver.

VQ systems are generally memory-intensive, but the memory requirements are symmetric with respect to the encoder and decoder. The codebook size is $\mathcal{O}(2^{RN})$ where R is the prescribed bit rate and N is the block size. This behavior, coupled with the fact that the codevector lengths obviously grow linearly with N, means that the codebook memory requirements grow dramatically with block size.

By contrast, the computational requirements of VQ systems are highly asymmetric. A full codebook search at the encoder has a computational complexity comparable

Gregory W. Wornell is also a consultant to PictureTel Corporation, Danvers, MA 01923.

1068-0314/96$5.00 © 1996 IEEE

to the memory requirements, viz., $\mathcal{O}(2^{RN})$ per sample. Decoding complexity is negligible however, since it requires a simple table lookup. This asymmetry is particularly well-suited to a variety of applications, such as database browsing. However, VQ systems and subsystems are also widely used in a wide spectrum of other applications, including videoconferencing and digital audio systems.

It is well known that VQ is an asymptotically optimal compression strategy—given a sufficiently long block length and suitably designed codebook, the rate-distortion bound for the source can be approached arbitrarily closely. However, the memory and computational requirements strongly limit block lengths, and as a result the asymptotic limits are rarely approached in practice. The use of constrained or structured codebooks can reduce the computational and/or memory requirements, allowing larger block sizes to be used [2] [3] [1]. However, with such constraints, VQ is generally no longer asymptotically optimal.

An important class of coding systems that can be interpreted as a form of VQ with constrained codebooks is the traditional approach of using a linear block transform followed by scalar quantization [4]. As is well known, the resulting system is equivalent to a VQ system in which the codebook corresponds to a rotated Cartesian lattice of codevectors. The memory requirements of such systems are dramatically reduced, to $\mathcal{O}(N2^R)$. Moreover, if a fast-computable transform is used, the computational complexity at both the encoder and decoder is only $\mathcal{O}(\log N)$ per sample. However, although reasonable performance can often be achieved via transform coding, its performance does not approach the rate-distortion bound with increasing block size.

1.1 Mitigation of Blocking Artifacts: Lapped VQ

The need to use finite block sizes in constrained and unconstrained VQ systems not only limits how closely the rate-distortion bound can be approached, but also leads to unnatural and perceptually distracting blocking artifacts. In effect, mean-square coding distortion is not minimized because interblock dependencies are not exploited, and blocking artifacts arise because the distortion that is introduced by the coding process has statistics that are periodic with a period equal to the block size.

One class of techniques for mitigating artifacts in block processing systems such as VQ involves applying a temporally- or spatially-varying filter to the reconstructed signal at the decoder [5] [6] [7]. Such techniques can be combined with suitable prefiltering to substantially reduce blocking artifacts, though at the expense of an increase in the overall mean-square reconstruction error [8] [9] [10].

More efficient and effective systems have generally resulted from the use of lapped block processing strategies. For example, in unconstrained (full-search) VQ systems, blocking artifacts can be reduced by extending the reconstruction codevectors beyond the block boundaries at the decoder. A mean-square optimized overlapping reconstruction codebook can lead to a noticeable reduction of blocking artifacts and a reduction of the reconstruction error [11]. However, a disadvantage of this particular approach is the increase in decoding complexity and memory requirements due to the increased decoder codebook size.

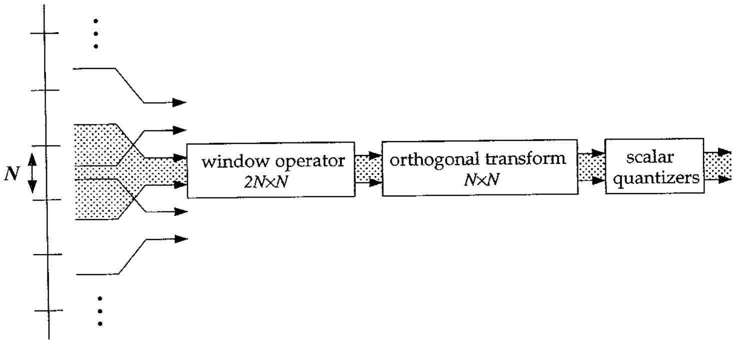


Figure 1: *Transform coding with lapped transforms. Typically, the quantizer block is formed from a set of N independent scalar quantizers.*

In the sequel, we develop an efficient lapped VQ scheme in which blocks are acquired in a lapped manner at the encoder, and reconstructed in a lapped manner at the decoder. As we will demonstrate, this technique produces performance enhancements similar to those in [11], but without requiring any increase in the coder or decoder codebook sizes. This new scheme can be interpreted as a powerful generalization of lapped transform coding schemes, whose relevant characteristics we summarize next.

1.2 Lapped Orthogonal Transforms

Lapped transform coding can be viewed as a lapped VQ strategy with a highly structured codebook, in much the same way as conventional transform coding can be viewed as a conventional VQ strategy with a highly structured codebook. Moreover, the use of lapped transforms with suitable orthogonality properties can achieve a significant reduction in blocking artifacts, and also simultaneously a reduction in mean-square reconstruction error over nonlapped transform coding.

With lapped transforms, the input signal is represented as a linear combination of overlapping basis functions. Although other sizes are also used in practice, often the basis functions from adjacent blocks overlap by 50% on each side of the block, so that their length is twice the block size [9] [12]. With such schemes, the transform matrix has size $N \times 2N$, mapping a block of $2N$ input samples into a block of N transform coefficients as shown in Fig. 1. Each length-$2N$ block in a lapped transform system cannot be exactly reconstructed from its N transform coefficients. However, when the transform basis functions satisfy the additional "orthogonality in the tails" constraint [12]—so that the collection of basis functions for all blocks constitute a complete orthonormal set—then, in the absence of quantization, perfect reconstruction of the signal can be achieved by superimposing the overlapped blocks at the decoder. These are referred to as lapped orthogonal transform (LOT) systems.

As Fig. 1 implies, the $2N \times N$ transform matrix $\mathbf{Q}$ of any LOT system can be (nonuniquely) factored into the product of a $2N \times N$ window operator matrix $\mathbf{W}$ and an $N \times N$ orthogonal transform matrix $\mathbf{U}$. For some LOT systems, this factorization can be performed so that $\mathbf{W}$ is a sparse matrix and $\mathbf{U}$ can be implemented via a fast algorithm [12]. For example, for the class of LOT systems referred to as modulated lapped transform (MLT) systems, the resulting $\mathbf{U}$ can be efficiently implemented via an $\mathcal{O}(\log N)$ per sample algorithm, and $\mathbf{W}$ via an algorithm whose complexity per sample is *independent* of block size, so that the overall complexity is $\mathcal{O}(\log N)$ per sample.

2 Lapped Orthogonal Vector Quantization

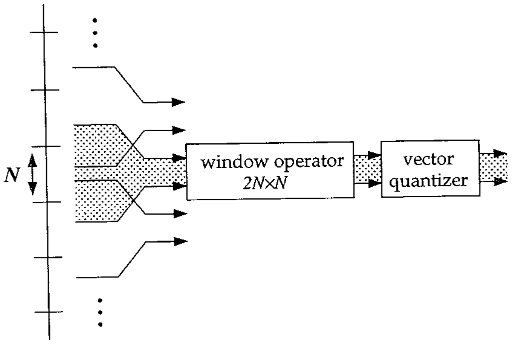
Efficient lapped VQ systems result from generalizing lapped transform coding systems. In lapped transform systems, the N transform coefficients generated for each block via the lapped transform are quantized via individual scalar quantizers. As a result, lapped transform coding corresponds to a lapped VQ strategy with a highly constrained codebook. In the remainder of this section, we focus on systems where the codebook is substantially less constrained. In particular, we replace the bank of N scalar quantizers in Fig. 1 with an unconstrained mean-square optimized vector quantizer whose codewords have length N. We refer to the result as lapped orthogonal vector quantization (LOVQ).

When VQ is used in place of the bank of scalar quantizers in the LOT structure of Fig. 1, the implementation of the lapped transform component of the system can be substantially simplified. In particular, the $N \times N$ orthogonal transform matrix $\mathbf{U}$ can be eliminated with no impact on performance; this matrix merely induces a (generalized) rotation of N-dimensional vector space, so its effect can be conveniently absorbed into the VQ subsystem design provided the VQ is unconstrained [1]. Note, however, that the window operator cannot be absorbed into the VQ since its dimension is $2N \times N$.

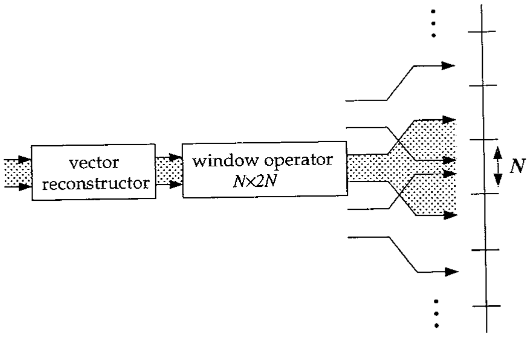
The resulting LOVQ encoder structure, which is equivalent to an LOT followed by VQ, is depicted in Fig. 2(a). The corresponding decoder structure, which is equivalent to a VQ decoder followed by the LOT inverse, is depicted in Fig. 2(b). The VQ decoder in Fig 2(b) is again a simple table lookup operation: the appropriate length-N codevector is selected according to the received index. The window operator inverse, in turn, maps successive length-N codevectors into overlapping length-$2N$ codevectors which are superimposed to generate the reconstruction at the output.

The choice of window operator has a significant impact on the performance of the resulting system, both in terms of mitigating blocking artifacts and reducing mean-square coding distortion. Furthermore, the structure of this operator affects the additional computational complexity inherent in the use of LOVQ over conventional VQ systems. From these perspectives, a particularly attractive choice for the window operator is that corresponding to the MLT. Its implementation via the orthogonal butterflies is depicted in Fig. 3, where the butterfly transmittances are given by $h[n] = \sin[(2n+1)\pi/4N]$.

As one might expect for a reconstruction that avoids blocking artifacts, this choice



(a) LOVQ encoding.



(b) LOVQ decoding.

Figure 2: *Implementation of lapped orthogonal vector quantization.*

for the window operator leads to the overlapping length-$2N$ decoder codevectors tapering smoothly to zero at both ends. This follows from the fact that each length-$2N$ codevector generated at the output of the LOVQ decoder in Fig. 2(b) is a linear combination of the N smoothly tapered basis functions of the window operator (which are the columns of the window operator matrix).

With the fast-computable MLT window, the LOVQ system complexity in terms of both computation and memory requirements is dominated by its VQ subsystem, and thus is comparable to that for traditional VQ systems. This makes LOVQ an attractive alternative to the lapped VQ scheme described in [11], which requires a decoder codebook whose vectors are of length $2N$.

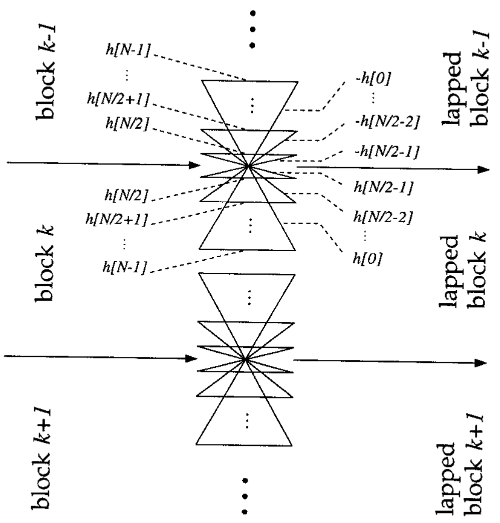


Figure 3: *LOVQ window operator at the encoder. The inverse window operator to be used at the decoder is the transposition of this signal flow graph.*

2.1 Optimization of Coding Gain in LOVQ Systems

Within the class of LOVQ systems, it is natural to seek that yielding both minimal block artifacts and minimal overall coding distortion. Fortunately, these objectives are nonconflicting. In this section we describe a framework for optimizing LOVQ systems.

To begin, first note that when the original source $x[n]$ is stationary, the sequence of overlapping length-$2N$ vectors $\mathbf{x}$ at the input to the window operator is a stationary vector source with Toeplitz covariance matrix $\mathbf{R_x}$. In turn, the length-N vectors of transform coefficients $\mathbf{y}$ at the input to the vector quantizer is also a stationary vector source with covariance matrix $\mathbf{R_y} = \mathbf{W}\mathbf{R_x}\mathbf{W}^{\mathrm{T}}$.

For ergodic sources, the mean-square distortion-rate function for blocks of size N is bounded according to $D_N(R) \leq \sigma_x^2\,\gamma_N^2\,2^{-2R}$, where σ_x^2 is the variance of the source, and where γ_N^2 is the spectral flatness measure for the source, i.e.,

$$\gamma_N^2 = \frac{\left[\prod\limits_{k=0}^{N-1} \lambda_k\right]^{1/N}}{\dfrac{1}{N}\sum\limits_{k=0}^{N-1} \lambda_k} = \frac{N\left[\det \mathbf{R_y}\right]^{1/N}}{\operatorname{tr} \mathbf{R_y}} = \frac{N\left[\det(\mathbf{W}\mathbf{R_x}\mathbf{W}^{\mathrm{T}})\right]^{1/N}}{\operatorname{tr}(\mathbf{W}\mathbf{R_x}\mathbf{W}^{\mathrm{T}})} \tag{1}$$

with λ_k denoting the kth eigenvalue of $\mathbf{R_y}$ [13].

The rate-distortion bound suggests that optimum VQ performance is obtained when the spectral flatness measure γ_N^2 is minimized. Thus, the desired optimization is to minimize (1) over all possible window operators $\mathbf{W}$ subject to the constraint that the operators correspond to orthogonal transformations. This constraint can be expressed in the form

$$\mathbf{W}\mathbf{W}^{\mathrm{T}} = \mathbf{I} \qquad \mathbf{W}\begin{bmatrix} 0 & \mathbf{I} \\ 0 & 0 \end{bmatrix}\mathbf{W}^{\mathrm{T}} = \mathbf{0}, \tag{2}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{0}$ is the zero matrix, both of size $N \times N$.

In addition, it is sometimes convenient to further constrain the window operator to have a fast implementation of the form described by Fig. 3. In this case, the orthogonality conditions (2) are equivalent to the condition that the window sequence $h[n]$ satisfy $h^2[n] + h^2[N-1-n] = 1$ for $n = 0, 1, \ldots, N/2 - 1$ [12].

Interestingly, for first-order autoregressive [AR(1)] sources $x[n]$, for which the autocorrelation function is $R_x[k] = \sigma^2 \rho^{|k|}$, the MLT window operator is asymptotically near-optimal, i.e., as $\rho \to 1$ except for very small block sizes. For $N = 2$, the optimal window sequence differs from that of the MLT, but can be readily computed, yielding

$$h[0] = \sin(\pi/6) \qquad h[1] = \cos(\pi/6). \tag{3}$$

3 LOVQ Performance Characteristics

Experiments involving speech and image data were conducted to verify the anticipated reduction in blocking artifacts. With the speech data, LOVQ based on the MLT window is compared with conventional VQ at rate $R = 0.5$ bits/sample and using a VQ block size of $N = 12$. Representative codevectors from the respective codebooks are depicted in Fig. 4. As the decoded waveform segments in Fig. 5 reflect, while traditional VQ led to both visibly and audibly significant blocking artifacts, these were effectively eliminated with LOVQ.

With the image data, LOVQ based on the MLT-window was compared to traditional VQ with 4×4 blocks ($N = 16$) at rate $R = 0.5$ bits/sample. Fig. 6 illustrates the performance of the respective systems on a test image of size 128×128 pixels, and 8 bits/pixel resolution. As Fig. 6 reflects, while using traditional VQ the reconstruction has prominent blocking artifacts, using LOVQ blocking effects are again effectively eliminated.

In both the above examples, the reduction of blocking artifacts was accompanied by a modest reduction in overall mean-square distortion as well. This byproduct is predicted by the theory described in Section 2.1. In particular, Fig. 7 depicts the coding gain that can be achieved for AR(1) sources using LOVQ with a fast window operator over conventional VQ with the same VQ block size N, as measured by the rate-distortion bound. For the case $N = 2$, the window operator that was used corresponded to the window sequence (3); for $N > 2$, the MLT window operator was used. Not surprisingly, greater coding gains are achieved for more strongly correlated sources and smaller block sizes.

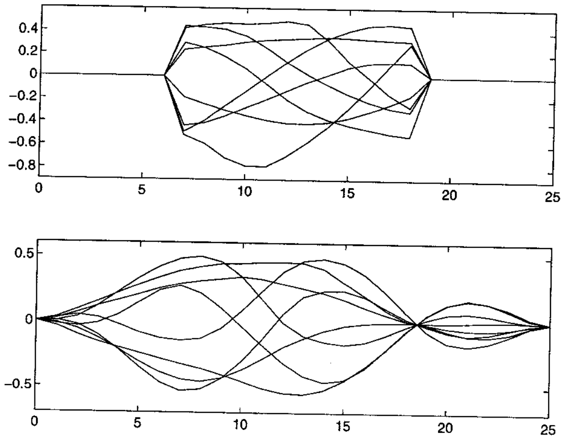


Figure 4: *Representative VQ codevectors for speech. Top: traditional VQ. Bottom: MLT-based LOVQ. Note the smooth decay of LOVQ codevectors, which reduce blocking artifacts.*

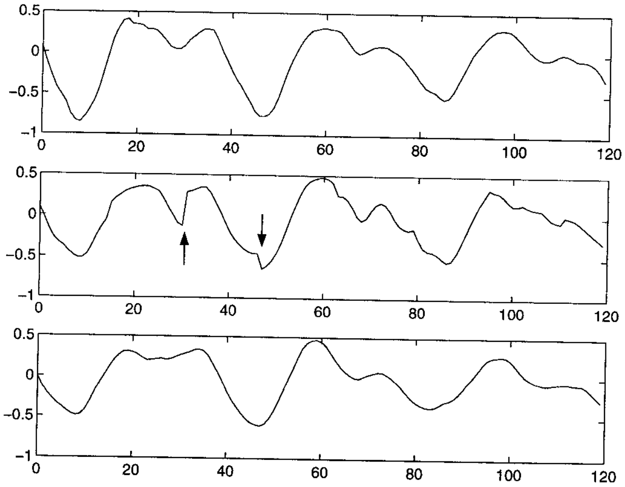


Figure 5: *Speech coding example at rate* $R = 0.5$ *bits/sample; VQ block size is* $N = 12$. *Top: original speech, sampled at 22.05 kHz. Middle: reconstruction with traditional VQ; the arrows indicate some of the prominent blocking artifacts. Bottom: reconstruction with LOVQ, which is effectively devoid of blocking artifacts.*

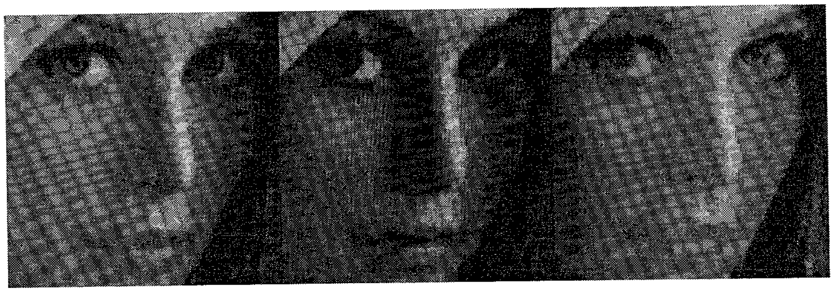


Figure 6: *Image coding example at rate $R = 0.5$ bits/sample; the block size is 4×4 ($N = 16$). Left: original image, 128×128 pixels, 8 bits/pixel resolution. Middle: reconstruction using traditional VQ. Right: reconstruction using LOVQ; note that blocking effects are mitigated without loss of resolution.*

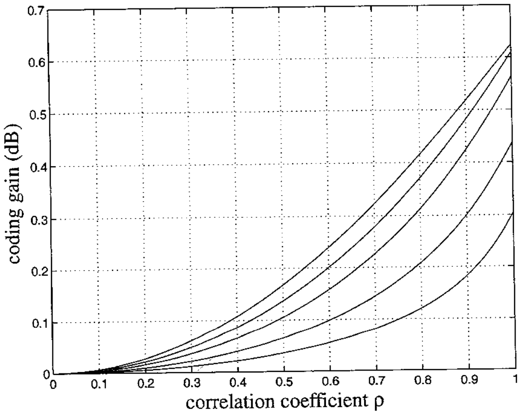


Figure 7: *Achievable coding gain of LOVQ over conventional VQ for an AR(1) source. The successively lower curves correspond to VQ block sizes $N = 2, 4, 8, 16, 32$.*

4 Conclusion

LOVQ has been developed as an efficient lapped VQ strategy that leads to dramatically reduced blocking artifacts when compared with traditional VQ systems. As an attractive byproduct, with LOVQ this reduction is also accompanied by a modest reduction in overall mean-square distortion. Most importantly, these performance enhancements are achieved with negligible increase in system complexity or memory requirements. In particular, the overhead in complexity amounts to a total of only 1.5 additional multiplies and adds per input sample.

References

[1] A. Gersho and R. Gray, *Vector Quantization and Signal Compression.* Boston, MA: Kluwer Academic Press, 1991.

[2] A. Buzo, A. H. Gray, Jr., R. M. Gray, and J. D. Markel, "Speech coding based upon vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-28, pp. 562–574, Oct. 1980.

[3] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices, and Groups.* Springer-Verlag, 1988.

[4] N. S. Jayant and P. Noll, *Digital Coding of Waveforms.* Englewood Cliffs, NJ: Prentice-Hall, 1984.

[5] H. C. Reeve, III and J. Lim, "Reduction of blocking effects in image coding," in *Proc. Int. Conf. Acoust. Speech, Signal Processing,* (Boston), pp. 1212–1215, 1983.

[6] B. Ramamurthi and A. Gersho, "Nonlinear space-invariant postprocessing of block coded images," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-34, pp. 1258–1268, Oct. 1986.

[7] X. Yuan, "Method and apparatus for processing block coded image data to reduce boundary artifacts between adjacent image blocks." U. S. Patent No. 5,367,385, Nov. 1994.

[8] H. S. Malvar, "Method and system for adapting a digitized signal processing system for block processing with minimal blocking artifacts." U. S. Patent No. 4,754,492, June 1988.

[9] H. S. Malvar, "The LOT: Transform coding without blocking effects," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 37, pp. 553–559, Apr. 1989.

[10] J.-C. Jeong, "Apparatus and method for encoding/decoding data including the suppression of blocking artifacts." U. S. Patent No. 5,384,849, Jan. 1995.

[11] S.-W. Wu and A. Gersho, "Lapped vector quantization of images," *Optical Engineering,* vol. 32, pp. 1489–1495, July 1993.

[12] H. S. Malvar, *Signal Processing with Lapped Transforms.* Norwood, MA: Artech House, 1992.

[13] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression.* Englewood Cliffs, NJ: Prentice-Hall, 1971.