Manipulating Rates, Complexity and Error-Resilience with Discrete Transforms

Vivek K Goyal[1,2] Martin Vetterli[2,3]

[1]Mathematics of Communications Research Department
Bell Labs, Lucent Technologies

[2]Dept. of Electrical Engineering and Computer Sciences
University of California, Berkeley

[3]Laboratoire de Communications Audiovisuelles
École Polytechnique Fédérale de Lausanne, Switzerland

E-mail: v.goyal@ieee.org, Martin.Vetterli@de.epfl.ch

Abstract

The common conception of transform coding is as a computationally efficient alternative to vector quantization. At high rates, it is not the partitioning itself but the efficiency of the scalar entropy coding which makes transform coding useful. With this view, a class of discrete transforms derived from linear transforms are used to pursue three objectives: reducing coefficient entropies (getting coding gain as in conventional transform coding), reducing the complexity of entropy coding (allowing many coefficients to be efficiently coded with identical entropy codes), and having robustness to coefficient erasures.

1. Introduction

Virtually all image, audio, and video coding standards use a structure consisting of a linear transform, scalar quantization, and entropy coding, in that order. Zero-tree structures [11] and similar developments mix the quantization and entropy coding to some degree, but it remains that the transform is calculated on continuous-valued (or "full precision") data.

Since the data will ultimately be represented coarsely, it seems that it should be sufficient to compute the transform coarsely.[1] Another approach that may reduce the complexity of the transform is to compute the transform on a discrete domain of about the same "size" as the final representation. Computing the transform on a "smaller" domain implies that the source is first quantized and then undergoes a transform.

[1]This was explored in a preliminary fashion in [5, 6].

This paper analyzes a few systems which combine—in the unusual specified order—scalar quantization, transform, and entropy coding. The use of discrete transforms provides more design freedom than we can handle. By restricting attention to a particular family of discrete transforms, we can describe forward and inverse transforms simply and follow principled design rules. The resulting systems provide opportunities for complexity reduction and reducing sensitivity to erasures.

The paper is organized as follows: Section 2 provides a review of transform coding. Sections 3–5 give the results on achieving coding gain, reduction in entropy-coding complexity, and robustness to erasures using a class of discrete transforms. This class of transforms is described in detail in Appendix A.

2. Transform Coding: A Brief Review

In its simplest incarnation, transform coding is the representation of a random vector $x \in \mathbb{R}^n$ by the following three steps:

- A transform coefficient vector is computed as $y = Tx$, $T \in \mathbb{R}^{n \times n}$.
- Each transform coefficient is quantized by a scalar quantizer: $\hat{y}_i = q_i(y_i)$, $i = 1, 2, \ldots, n$. The overall quantizer is denoted $Q : \mathbb{R}^n \to \mathbb{R}^n$.
- An entropy code is applied to each quantized coefficient: $E_i(\hat{y}_i)$, $i = 1, 2, \ldots, n$.

The decoder reverses the steps to produce an estimate $\hat{x}$. The mean-squared error distortion of the scheme is $n^{-1}\mathbf{E}\|x - \hat{x}\|_2^2$, where $\mathbf{E}$ is the expectation operator. The rate is $n^{-1}\mathbf{E}[\sum_i \ell(E_i(\hat{y}_i))]$, where $\ell(\cdot)$ gives the length of a codeword.

We will consider only the coding of i.i.d. jointly Gaussian sources. Huang and Schultheiss [8] considered the optimal design of the transform when Lloyd-Max (optimal fixed-rate) quantizers are used. Under a mild condition on the bit allocation, they showed that the transform should be a Karhunen-Loève transform (KLT) of the source; *i.e.*, a transform that produces uncorrelated transform coefficients.

Using high rate approximations, it is easy to extend the optimality of the KLT to entropy-coded unbounded uniform quantization [3]. An (unbounded) uniform quantizer with step size Δ maps an input variable to the nearest multiple of Δ. This type of quantization will be assumed throughout the paper and will be denoted $[\cdot]_\Delta$. At high rates (small Δ), the quantization error is uniform and the distortion is given by

$$D = \Delta^2/12. \tag{1}$$

Assuming ideal entropy coding, the rate for quantizing a Gaussian random variable with variance σ^2 is a function of Δ and the differential entropy of the source [2]:

$$R \approx \frac{1}{2}\log_2 2\pi e\sigma^2 - \log_2 \Delta.$$

The average rate for the n transform coefficients is

$$R \approx \frac{1}{2}\log_2 2\pi e\Big(\prod_{i=1}^{n}\sigma_{y_i}^2\Big)^{1/n} - \log_2 \Delta. \tag{2}$$

Combining (1) and (2) yields

$$D \approx \frac{\pi e}{6}\Big(\prod_{i=1}^{n}\sigma_{y_i}^2\Big)^{1/n} 2^{-2R}. \tag{3}$$

Why does transform coding work? An algebraic answer is that the transform makes $\prod \sigma_{y_i}^2 < \prod \sigma_{x_i}^2$, but this is not very enlightening. The geometry of the situation is shown in Figure 1. The ellipses represent level curves of the p.d.f. of the source. Quantization in the original coordinates is shown on the left, and quantization after the KLT is shown on the right. Is the second partition any better than the first? Put in another way, is $Q(T(\cdot))$ a better vector quantization encoder mapping than $Q(\cdot)$? In the high rate limit, the answer is no, since either gives distortion $D = \Delta^2/12$. Transform coding does not improve the quantization, but rather makes the *scalar* entropy coding that follows it work well; if the entropy coding processed an entire vector at a time, the transform would give no advantage.[2]

Since the quantization performance is not effected much by the transform, we may try to replace the "T-Q-E" structure of transform-quantization-entropy coding with a "Q-T-E" structure. Considering only linear transforms from $\Delta\mathbb{Z}^n$ to $\Delta\mathbb{Z}^n$ would be too restrictive, but placing no restriction on the transform gives more design freedom than we can deal with well. Thus, to have easily implemented transforms and to simplify the design process we use only transforms which are derived from (continuous) linear transforms, as described in Appendix A.

Discrete domain transforms can nearly achieve the coding gain of traditional transform coding, but at the same time introduce other possibilities. Though the remainder of the paper addresses the coding of Gaussian sources, the use of discrete transforms extends the applicability of transform coding to discrete sources, and perhaps to abstract alphabet (non-numerical) sources. For simplicity most expressions and all simulations are for two-dimensional sources.

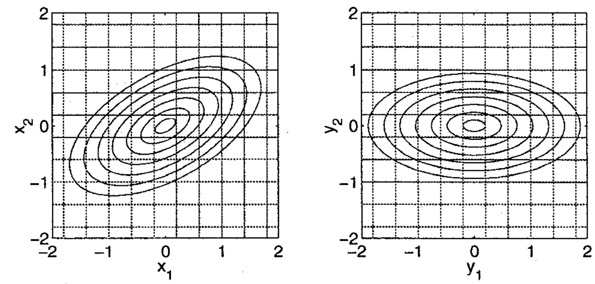


Figure 1. Partitioning induced by uniform scalar quantization with and without the application of the Karhunen-Loève transform. The ellipses are level curves of the p.d.f. of the source.

3. Rate Reduction

In transform coding with quantization preceding transform, the distortion is fixed (at approximately $\Delta^2/12$) independent of the transform. The role of the transform is to reduce the coded rate, but an invertible transform cannot effect the entropy for a discrete random variable [2]. This seeming contradiction is resolved by again remembering that we wish for the entropy coding to operate on scalars.

Denote the source by (x_1, x_2), the transform by $\hat{T}$, and the transform coefficients by $(y_1, y_2) = \hat{T}(x_1, x_2)$. In the best case scenario, y_1 and y_2 are independent, so we take advantage of

$$H(x_1) + H(x_2) \geq H(x_1, x_2) = H(y_1, y_2) = H(y_1) + H(y_2),$$

where the left hand side is a lower bound on the rate without the transform. We cannot normally expect to make y_1 and y_2 independent, but we can approximately achieve this condition by choosing $\hat{T}$ to be an approximation to a KLT

[2]We are concerned here with high rates; at low rates it is hard to predict the best coordinates for scalar quantization.

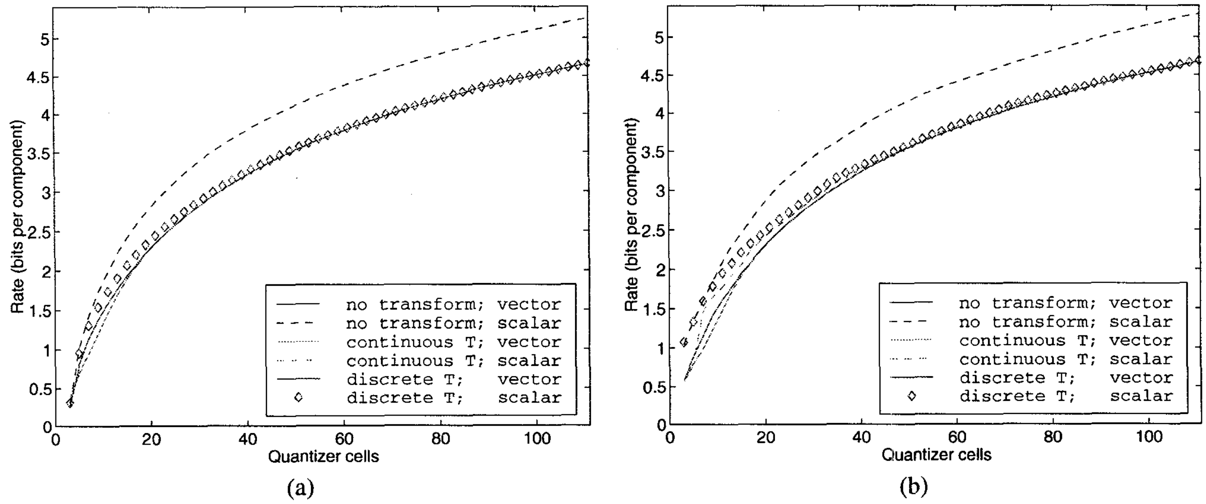


Figure 2. Experiment showing that the coding gain of the (continuous) KLT is almost matched by a discrete transform which approximates the KLT. The horizontal axis gives the number of cells covering $[-6, 6]$ for each component quantizer. Legend entries indicate whether no transform, a KLT, or a discrete approximation of the KLT was used; and whether the entropy coding is based on scalars or vectors. (a) Rates based on empirical entropies; (b) Rates based on explicit Huffman codes.

for x. Because the construction of the discrete transform introduces only $O(\Delta)$ error (see Appendix A), in the high rate limit y_1 and y_2 are independent.

This was experimentally confirmed with a two-dimensional Gaussian source with correlation matrix

$$R_x = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}.$$

The KLT is a $\pi/4$ radian rotation. A comparison between using no transform, using the KLT (before quantization), and using a discrete approximation to the KLT is shown in Figure 2. If the entropy coding operates on vectors there is virtually no difference between the three transform choices.[3] Removing the correlation in the source is important with scalar entropy coding. The discrete transform performs almost as well as the KLT; of course, it cannot perform better because the KLT makes the transform coefficients independent.

4. Complexity Reduction

The previous section demonstrated that a discrete transform can do about as well as a continuous transform when scalar entropy coding is to be used. It was implicit that each scalar entropy code was optimized to its corresponding transform coefficient. Having n separate entropy codes increases the memory requirements and is thus undesirable. In the previous example, this could be seen as an argument for using scalar entropy coding in the original coordinates, despite the higher rate.

For simplicity, consider a source with independent components: $R_x = \operatorname{diag}(\sigma_{x_1}^2, \sigma_{x_2}^2)$, $\sigma_{x_1} \geq \sigma_{x_2}$. There is little flexibility in the choice of continuous transforms since they must be orthogonal to maintain cubic partition cells. For this source, only trivial rotations of $k\pi/2$ radians do not increase the rate (see (2)); thus, there is no hope to equalize the p.d.f.'s of the transform coefficients without hurting the rate-distortion performance. The family of discrete transforms used here gives more flexibility. We need not start with an orthogonal transform; any transform with determinant 1 will suffice. Discrete transforms derived from initial transforms of the form

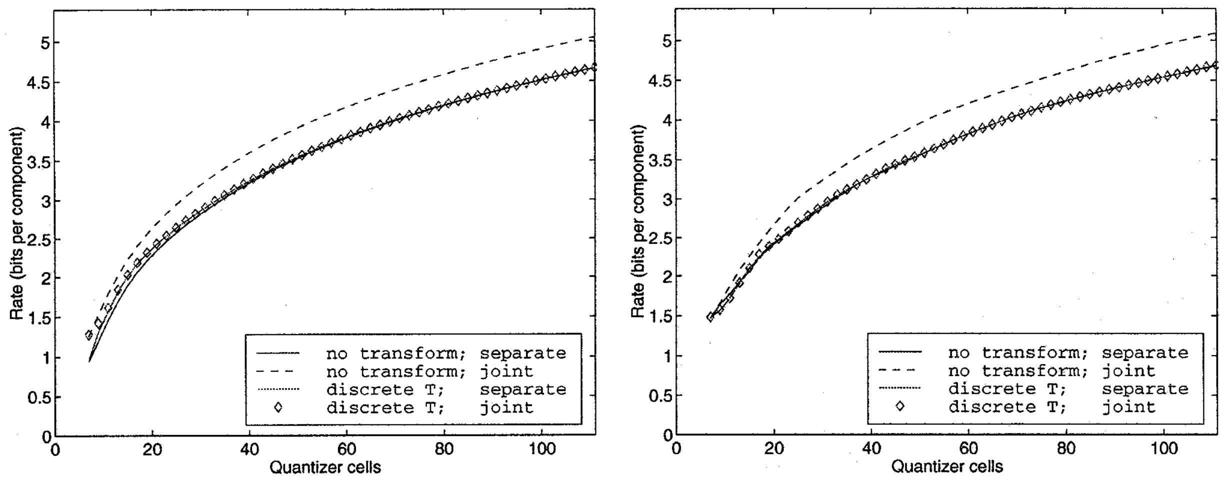
$$\hat{T} = \begin{bmatrix} \alpha & \pm\sigma_{x_2}^{-1}\alpha\sigma_{x_1} \\ \mp(2\alpha\sigma_{x_1})^{-1}\sigma_{x_2} & (2\alpha)^{-1} \end{bmatrix}$$

all give the optimal coding gain. In particular,

$$\hat{T} = \begin{bmatrix} \alpha & (2\alpha)^{-1} \\ -\alpha & (2\alpha)^{-1} \end{bmatrix} \text{ with } \alpha = \sqrt{\frac{\sigma_{x_2}}{2\sigma_{x_1}}} \tag{4}$$

gives optimal coding gain and produces transform coefficients with identical distributions. See [4] for a general analysis of which this is a special case.

[3] The "no transform; vector" and "discrete T; vector" cases give precisely the same rates, as do "continuous T; vector" and "continuous T; scalar."



(a) (b)

Figure 3. Experiment showing that a discrete transform makes it possible to simultaneously achieve optimal coding gain and use the same entropy code for each transform coefficient. The horizontal axis gives the number of cells covering $[-6, 6]$ for each component quantizer. Legend entries indicate whether a transform is used and whether the separate entropy codes are used for each transform coefficient. (a) Rates based on empirical entropies; (b) Rates based on explicit Huffman codes.

An experimental confirmation is shown in Figure 3. The source was chosen to have the same power and eigenvalue spread as in the previous example, so $\sigma_{x_1}^2 = 1.9$ and $\sigma_{x_2}^2 = 0.1$. Since the source components are independent, the best case performance is to quantize and apply separately optimized entropy codes to the two variables. However, when a discrete transform based on (4) is used, the best performance is almost matched even with a single entropy code applied to both transform coefficients.

Other manipulations of the transform coefficient variances may be useful. For example, the probability of a zero coefficient effects both the efficacy of run-length coding and decoding optimizations in the spirit of [9].

5. Erasure Resilience

The choice of α in (4) is the extreme case of an analysis in [4]. Transforms of that form with larger values of α increase the rate, but in return improve the ability to reconstruct from only one of the two transform coefficients. This is useful in what is called multiple description coding.

The effect can be understood with reference to R_y. The product of the diagonal elements of R_y is increased, which according to (2) increases the rate. At the same time, the off-diagonal elements of R_y are no longer zero. This means that the transform coefficients are correlated, and if one is lost it can be estimated from the other [4, 10].

A. Pseudo-linear Discrete Transforms

Recently, several researchers have proposed using invertible discrete-domain to discrete-domain transforms [1, 7, 13]. They appear under various names (lossless transforms, integer-to-integer transforms, lifting factorizations) and in various flavors (finite dimensional matrices, or Fourier or wavelet domain operators). All these transforms are based on factorizations of matrices which make information flow in a simple, regular way. Inversion can then be achieved by reversing the information flow.

For example, one can factor any 2×2 matrix with determinant 1 into three lower- and upper-triangular matrices with unit diagonals:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 \\ \frac{d-1}{b} & 1 \end{bmatrix}}_{T_1} \underbrace{\begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix}}_{T_2} \underbrace{\begin{bmatrix} 1 & 0 \\ \frac{a-1}{b} & 1 \end{bmatrix}}_{T_3} \text{ or}$$
$$= \underbrace{\begin{bmatrix} 1 & \frac{a-1}{c} \\ 0 & 1 \end{bmatrix}}_{T_1} \underbrace{\begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix}}_{T_2} \underbrace{\begin{bmatrix} 1 & \frac{d-1}{c} \\ 0 & 1 \end{bmatrix}}_{T_3}. \quad (5)$$

Since the inverse of a block $\begin{bmatrix} 1 & 0 \\ x & 1 \end{bmatrix}$ or $\begin{bmatrix} 1 & y \\ 0 & 1 \end{bmatrix}$ is simply $\begin{bmatrix} 1 & 0 \\ -x & 1 \end{bmatrix}$ or $\begin{bmatrix} 1 & -y \\ 0 & 1 \end{bmatrix}$, respectively, the inverse of (5) can be found by reversing the order of the factors and changing the signs of the off-diagonal elements.

The more profound fact is that the simplicity of inversion remains if the off-diagonal elements represent nonlinear functions. Let $[\,\cdot\,]_\Delta$ represent rounding to the nearest multiple of Δ and let

$$T_1 = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}.$$

If $x \in \Delta\mathbb{Z}^2$, then

$$[T_1 x]_\Delta = \left[\begin{bmatrix} x_1 + ax_2 \\ x_2 \end{bmatrix}\right]_\Delta = \begin{bmatrix} x_1 + [ax_2]_\Delta \\ x_2 \end{bmatrix}.$$

Thus $[T_1 \cdot]_\Delta$ is an identity operator except for a nonlinear function of x_2 being added to x_1. Direct computation shows that *on the domain* $\Delta\mathbb{Z}^2$, $[T_1^{-1} \cdot]_\Delta$ is the inverse operator. A cascade of such operations is invertible in the same manner, so a factorization $T = T_1 T_2 T_3$ yields an invertible discrete transform $\hat{T} : \Delta\mathbb{Z}^2 \to \Delta\mathbb{Z}^2$ "derived from T" through

$$\hat{T}(x) = \left[T_1 \left[T_2 \left[T_3 x\right]_\Delta\right]_\Delta\right]_\Delta. \tag{6}$$

The discrete transform $\hat{T}$ depends not only T, but the factorization of T. Among the possible factorizations, one can minimize a bound on $\|\hat{T}(x) - Tx\|$. Let

$$T_1 = \begin{bmatrix} 1 & 0 \\ a & 1 \end{bmatrix},\ T_2 = \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix},\ \text{and } T_3 = \begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix}.$$

For $x \in \Delta\mathbb{Z}^2$, the computation (6) involves three rounding operations. Using δ_i's to denote the roundoff errors gives

$$\hat{T}(x) = T_1 \left(T_2 \left(T_3 x + \begin{bmatrix} 0 \\ \delta_1 \end{bmatrix}\right) + \begin{bmatrix} \delta_2 \\ 0 \end{bmatrix}\right) + \begin{bmatrix} 0 \\ \delta_3 \end{bmatrix}.$$

Expanding and using $T_1 T_2 T_3 = T$, one can compute

$$\left\|\hat{T}(x) - Tx\right\|_\infty \ \leq\ (1 + \max\{|b|, |a| + |1 + ab|\})\,\frac{\Delta}{2}.$$

This shows that $\hat{T}$ approximates T in a precise sense; in particular, $\hat{T}(x) \approx Tx$ when Δ is small.

For $n \times n$ matrices, the process is similar. T is factored into a product of matrices with unit diagonals and nonzero off-diagonal elements only in one row or column: $T = T_1 T_2 \cdots T_k$. The discrete version of the transform is then given by

$$\hat{T}(x) = \left[T_1 \left[T_2 \ldots \left[T_k x\right]_\Delta\right]_\Delta\right]_\Delta.$$

The lifting structure ensures that the inverse of $\hat{T}$ can be implemented by reversing the calculations.

The existence of such a factorization follows from the fact that any nonsingular matrix can be reduced to an identity matrix by multiplication with elementary matrices [12]. Since our original matrix has determinant 1, it is sufficient to consider the following three types of elementary matrices:

- $E_{ij}^{(\lambda)}$, to subtract a multiple λ of row j from row i.
- P_{ij}, to exchange rows i and j.
- $D_{ij}^{(\lambda)}$, to multiply row i by λ and row j by $1/\lambda$.

$E_{ij}^{(\lambda)}$ is already in the desired form. The remaining two can be factored as desired using the factorization of 2×2 matrices above.

References

[1] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo. Wavelet transforms that map integers to integers. *Appl. Computational Harmonic Anal.*, 5(3):332–369, July 1998.

[2] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, 1991.

[3] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Acad. Pub., Boston, 1992.

[4] V. K. Goyal and J. Kovačević. Optimal multiple description transform coding of Gaussian vectors. In *Proc. IEEE Data Compression Conf.*, pp. 388–397, Snowbird, Utah, Mar. 1998.

[5] V. K. Goyal and M. Vetterli. Computation-distortion characteristics of block transform coding. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, vol. 4, pp. 2729–2732, Munich, Germany, Apr. 1997.

[6] V. K. Goyal and M. Vetterli. Computation-distortion characteristics of JPEG encoding and decoding. In *Proc. 31st Asilomar Conf. Sig., Sys., & Computers*, Pacific Grove, CA, Nov. 1997.

[7] J. Hong. *Discrete Fourier, Hartley, and Cosine Transforms in Signal Processing*. PhD thesis, Columbia Univ., 1993.

[8] J. J. Y. Huang and P. M. Schultheiss. Block quantization of correlated Gaussian random variables. *IEEE Trans. Comm.*, 11:289–296, Sept. 1963.

[9] K. Lengwehasatit and A. Ortega. Distortion/decoding time tradeoffs in software DCT-based image coding. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, vol. 4, pp. 2725–2728, Munich, Germany, Apr. 1997.

[10] M. T. Orchard, Y. Wang, V. Vaishampayan, and A. R. Reibman. Redundancy rate-distortion analysis of multiple description coding using pairwise correlating transforms. In *Proc. IEEE Int. Conf. Image Proc.*, vol. I, pp. 608–611, Santa Barbara, Oct. 1997.

[11] J. M. Shapiro. Embedded image coding using zero trees of wavelet coefficients. *IEEE Trans. Signal Proc.*, 41(12):3445–3462, Dec. 1993.

[12] G. Strang. *Linear Algebra and Its Applications, Third Edition*. Harcourt Brace Jovanovich, San Diego, 1988.

[13] A. Zandi, J. D. Allen, E. L. Schwartz, and M. Boliek. CREW: Compression with reversible embedded wavelets. In *Proc. IEEE Data Compression Conf.*, pp. 212–221, Snowbird, Utah, Mar. 1995.