# Universal Transform Coding Based On Backward Adaptation

Vivek K Goyal, Jun Zhuang, and Martin Vetterli*
Dept. of Electrical Engineering and Computer Sciences
University of California, Berkeley
{vkgoyal,zhuang,martin}@eecs.berkeley.edu
http://www-wavelet.eecs.berkeley.edu/

**Abstract**

The method for universal transform coding based on backward adaptation introduced in [1] is reviewed and further analyzed. This algorithm uses a linear transform which is periodically updated based on a local Karhunen-Loève Transform (KLT) estimate. The KLT estimate is derived purely from quantized data, so the decoder can track the encoder state without any side information. The effect of estimating only from quantized data is quantitatively analyzed. Two convergence results which hold in the absence of estimation noise are presented. The first applies for any vector dimension but does not preclude the necessity of a sequence of quantization step sizes that goes to zero. The second applies only in the two-dimensional case, but shows local convergence for a fixed, sufficiently small quantization step size. Refinements which reduce the storage and computational requirements of the algorithm are suggested.

## 1 Introduction

Universal source coding is the coding of a source with an unknown distribution. In universal lossless source coding, there are several practical methods which are capable of approaching the entropy rate bound. Universal codes for lossy source coding which approach the rate-distortion bound have been proven to exist but are not practical. This is because approaching the rate-distortion bound (with or without knowledge of the source distribution) requires increasing the quantizer vector dimension without bound. Since the computation associated with using a vector quantizer increases exponentially with the vector dimension, practical coding methods must place relative low limits on the vector dimension.

*J. Zhuang is currently with Pacific Bell, San Ramon, CA. M. Vetterli is also with École Polytechnique Fédérale de Lausanne, Switzerland.

1068-0314/97$10.00 © 1997 IEEE

Transform coding (with scalar quantization) provides a compromise between computational complexity and performance. Limiting the quantization to scalars makes feasible the use of longer vectors than would be possible had vector quantization been used. Coding of longer vectors facilitates the exploitation of linear intersample dependencies. The primary limitations of transform coding are that it can not exploit nonlinear dependencies and it does not benefit from packing gain.

In transform coding of a stationary source with a known distribution, it is well known that, subject to certain conditions, the optimal transform is the data-dependent Karhunen-Loève Transform (KLT). The KLT is the unique transform that both decorrelates the data (the transformed data has uncorrelated components) and gives optimal energy compaction (the diagonal entries of the correlation matrix of the transformed data are in decreasing order).

The study of adaptive transform coding has been a rich research area for several years. Recently, very good transform coding results have been reported using classification based methods, *i.e.* schemes in which the signal space is divided into a finite set of classes and a fixed transform is designed for each class [2, 3]. Classification methods generally rely on training for defining classes and/or designing a transform code for each class.

This paper extends results on a method for universal transform coding that we introduced in [1]. This scheme does not rely on classification or on *a priori* training; instead, it periodically adjusts the transform to approximately match the KLT estimated from local statistics. The novelty of this method is that the adaptation depends only on quantized data and hence the encoder and decoder can maintain the same state without any side information. The coder operates with a fixed vector dimension and uses only scalar quantization; thus, it can not be expected to perform at the rate-distortion bound. The goal is to achieve the performance of an optimal transform coder despite an initial lack of knowledge about the source distribution.

The earlier paper [1] described the feasibility of this backward adaptive scheme and presented experimental results on the coding of Gauss-Markov sources. The present paper discusses the possibilities and limitations in backward adaptive coding (Section 3), establishes convergence properties of the algorithm (Section 4), and suggests computational refinements (Section 5).

## 2 Structure of the Proposed Coder

An intuitive requirement for an encoder to achieve universality is for it to "learn" the statistics of the unknown or time-varying source. Given a block of data, the standard way to make a coding system adaptive is to use the data (and perhaps prior blocks) to develop a model of the source and design an encoder optimal for the source prior to coding the data. In order for the receiver to correctly decode the data, it must be informed of the adaptation of the encoder; hence the parameters of the encoder are sent along with the coded data. The main sources of performance degradation compared to an optimal "omniscient" source coder are modeling error and the cost of sending encoder parameters to the receiver. A block diagram for such a system

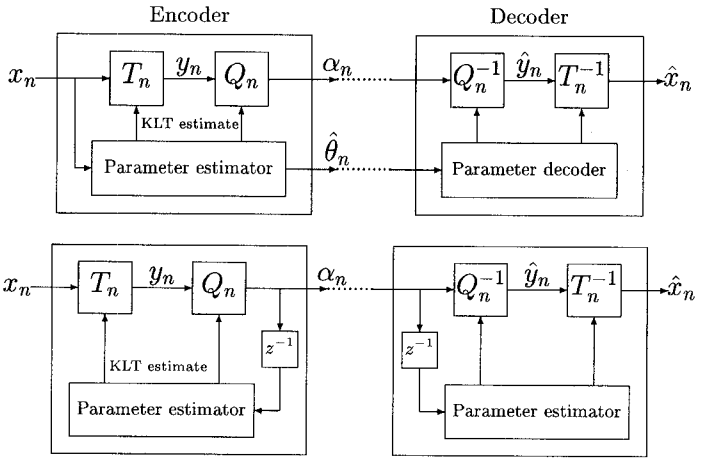


Figure 1: Structural comparison between a typical forward adaptive transform coding system (top) and the proposed backward adaptive system (bottom).

is shown in the top half of Figure 1. The sequence of source vectors is represented by $\{x_n\}_{n=0}^{\infty} \subset \mathbb{R}^N$, where we usually but not always assume that $x_n$ is formed by blocking a scalar-valued source. The linear transform and scalar quantizer used for vector $x_n$ are denoted by $T_n$ and $Q_n$, respectively; the subscript emphasizes that these are potentially adapted.

This work asks, "How well can one do *without* sending any encoder parameters to the receiver?" For the receiver to be able to correctly decode the data without being explicitly informed of the encoder state, it is necessary that all encoder adaptation depend only on information already available at the decoder at no additional cost, *i.e.* the *coded* data stream. In the parlance of communication, we are avoiding *side information* by using *backward adaptation*.[1] With this strategy, one avoids the price of side information but the efficiency of estimating source statistics is reduced. Also, there is some performance penalty from the requirement that adaptation be *strictly* causal. A block diagram of the proposed coder is shown in the bottom half of Figure 1. Structurally, eliminating the need for side information is much like put the quantization "inside the loop" in ADPCM.

# 3 Transform Adaptation

For an arbitrary source distribution and bit rate, the KLT is not necessarily optimal. However, because of a lack of methods for finding an optimal transform that are

[1] In practical forward adaptive transform coding systems in which only the quantization is adapted, 20 to 40 percent of the available bit rate is assigned to side information [4, §2.3]. A system which adapts the transform itself would presumably require at least as much side information.

practical for "on-line" use, we will in fact attempt to estimate and use the KLT. In particular, we will assume that the transform we wish to calculate is the local KLT, *i.e.* the KLT assuming the signal is wide-sense stationary with autocorrelation equal to that measured locally.

Determining the local KLT requires estimates of the second order statistics of $x_n$, namely $R_{x_n} = E[x_n x_n^T]$. If the unquantized signal $x_n$ could be used in parameter estimation, we could replace the ensemble average with a time average. The following subsections address the estimation of $R_{x_n}$ when only the quantized version $\hat{x}_n$ can be observed.

## 3.1 Parametric case

Consider the coding of a scalar source and suppose that the source can be described by a parametric model. Then, as described by Yu [5], the parameters of the source can in general be consistently estimated from observations of a quantized version of the source as long as the number of quantization bins exceeds the number of parameters. A scalar quantized random vector could be treated very similarly, with bins that are cartesian products of the bins in the scalar case.[2] After finding the parameters describing the source, one can find the moments needed to calculate the KLT.

The approach described above is not entirely satisfactory because it requires the estimation of a large number of bin probabilities.[3] In the case of a Gaussian source and uniform quantization, the situation is simpler because the parameters of the unquantized signal can be estimated from just the *moments* of the quantized signal as opposed to all of the relative bin probabilities. This is made more precise by the following theorem [1]:

**Theorem 1** *Let* $X = [X_1, \ldots, X_k]^T$, $X \sim \mathcal{N}(0, \Sigma)$, *where* $\Sigma$ *is an unknown non-degenerate covariance matrix. Let* $\hat{X}$ *be a scalar quantized version of* $X$ *such that for* $n \in \mathbb{Z}$ *either*

*(i)* $X_i \in [n\Delta_i, (n+1)\Delta_i) \Rightarrow \hat{X}_i = (n + \frac{1}{2})\Delta_i$; *or*

*(ii)* $X_i \in [(n - \frac{1}{2})\Delta_i, (n + \frac{1}{2})\Delta_i) \Rightarrow \hat{X}_i = n\Delta_i$.

*Then for any set of positive, finite quantization step sizes* $\Delta_1, \ldots, \Delta_k$, *all moments of* $X$ *can be recovered exactly from the first and second order moments of* $\hat{X}$.

The proof is based on finding the mapping between the moments of $X$ and the moments of $\hat{X}$ and then showing that this mapping is invertible. The general form of the mapping is complicated, but if the quantization step sizes are small very simple approximations can be used [1].

[2] It is interesting to note that even very coarse scalar quantization can yield enough information to fit a reasonable parametric model. For example, quantizing with only three bins will yield $3^k - 1$ independent probability estimates, where $k$ is the vector dimension. For any $k \in \mathbb{Z}^+$, $3^k - 1 \geq \frac{1}{2}k^2 + \frac{3}{2}k$, so this quantization is fine enough to fit a multivariate Gaussian signal model.

[3] The number of bins is exponential in the vector dimension.

## 3.2 Estimation noise

Since quantization is an irreversible reduction in information, it must be at least as hard to estimate the moments of a signal from a quantized version as it is from the original unquantized signal. This subsection quantifies this chief disadvantage of a backward adaptive system. Theorem 1 shows constructively how the moments of the unquantized signal can be recovered from the moments of the quantized signal. However, this alone does not tell the whole story: the moments of the quantized signal must also be estimated.

Let $X_1, X_2, \ldots, X_k$ be an i.i.d. sequence of Gaussian random variables with mean zero and unknown variance $\sigma^2$. It is easy to check that the sample variance $s^2 = \frac{1}{k}\sum_{i=1}^{k} X_i^2$ is an unbiased estimator of the variance.[4] The variance of this estimate is given by $E[(s^2 - \sigma^2)^2] = \frac{2\sigma^4}{k}$.

Now suppose that instead of observing $X_1, X_2, \ldots, X_k$, we observe quantized values $\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_k$, quantized as in case (ii) of Theorem 1. To estimate $\sigma^2$, we can first estimate $\hat{\sigma}^2 \equiv E[\hat{X}_1^2]$ and then invert the mapping which relates $\sigma^2$ and $\hat{\sigma}^2$. The quality of the estimate thusly obtained depends on the quality of the estimate of $\hat{\sigma}^2$ (the variance of the sample variance $\hat{s}^2 = \frac{1}{k}\sum_{i=1}^{k} \hat{X}_i^2$) and the sensitivity of the relationship between $\sigma^2$ and $\hat{\sigma}^2$ to errors in $\hat{\sigma}^2$. Using a first order approximation, we obtain

$$\mathrm{Var}(\sigma^2 \text{ estimate}) \approx \frac{\partial[\sigma^2]}{\partial[\hat{\sigma}^2]} \cdot \mathrm{Var}(\hat{s}^2). \tag{1}$$

An elementary calculation shows that $\mathrm{Var}(\hat{s}^2) = \frac{E[\hat{X}_1^4] - E[\hat{X}_1^2]}{k}$, where one can obtain expressions for $E[\hat{X}_1^4]$ and $E[\hat{X}_1^2]$ by manipulating expressions from [6]. Normalizing the variance of the estimate of $\sigma^2$ (approximated through (1)) by $\frac{2\sigma^4}{k}$ (the variance obtained without quantization) characterizes precisely how much is lost by estimating from quantized data. For example, if the quantization is such that $\Delta/\sigma = 3$ (quite coarse), one needs about twice as much data to estimate $\sigma^2$ as well as if the unquantized data were available. A similar analysis can be done for covariance estimates.

## 3.3 Nonparametric case

If a parametric source model is not known, making corrections as in Theorem 1 is simply impossible. An open theoretical question is to determine the effect of such corrections if an incorrect source model is used. Nevertheless, since a source model can generally not be guaranteed, for the remainder of the paper it can be assumed that estimates of the moments of the quantized signal are used directly as estimates of the moments of the original unquantized signal.

[4] The sum is divided by $k$ because the mean is known; if the mean was unknown and the variance was estimated by summing the squares of the deviations from the *sample* mean, dividing by $k - 1$ would give an unbiased estimator.

# 4 Convergence Results

The results of §3.1 indicate that if a parametric source model is known, in the absence of estimation noise[5] (or asymptotically as the estimation noise is driven to zero), the optimal transform can be computed. But as mentioned in the previous section, the actual proposed algorithm uses estimates of moments of the quantized signal *directly* as estimates of the moments of the original signal. In a variety of simulations, some of which were reported in [1], this algorithm has converged. In analytically studying the convergence of the algorithm, it is convenient to first consider a "non-stochastic" version of the algorithm where certain quantities which would be estimated online are replaced by their expected values. This non-stochastic version is analyzed in this section.

## 4.1 General convergence result

Let $x \in \mathbb{R}^N$ be a random vector with $R_x = E[xx^T]$. Let $T_0 \in \mathbb{R}^{N\times N}$ be an orthonormal matrix. Consider an iteration which updates the transform as follows:

- $y_n = T_n x$, so $R_{y_n} = E[y_n y_n^T] = T_n R_x T_n^T$.
- $\hat{y}_n = q_n(y_n)$, where $q_n$ is a uniform scalar quantizer with stepsize $\Delta_n$. Denote the effect of the quantization on the autocorrelation by $R_{\hat{y}_n} = Q(R_{y_n})$.
- $\hat{x}_n = T_n^T \hat{y}_n$, so $R_{\hat{x}_n} = T_n^T R_{\hat{y}_n} T_n$.
- $T_{n+1}$ is the KLT of $R_{\hat{x}_n}$, so $R_{\hat{x}_n} = T_{n+1}^T \Lambda_n T_{n+1}$, where $\Lambda$ is a diagonal matrix.

The function $Q$ depends on the distribution of $x$. Theorem 2 shows that under certain conditions on $Q$, there exists a sequence of step sizes such that this deterministic iteration converges to a transform which decorrelates $x$, *i.e.* $R_{y_n}$ approaches a diagonal matrix. In order to measure the degree to which $T_n$ decorrelates $x$, we define a distance measure $|||\cdot|||$ between a matrix $A$ and the set of diagonal matrices by $|||A||| = \sum_{i\neq j} a_{ij}^2$.

**Theorem 2** *Suppose $Q$ is such that $Q(A)$ is diagonal if and only if $A$ is diagonal. Suppose also that $Q(A)$ can be written as $Q(A) = A + bI_N + C$, where $b \in \mathbb{R}$, $C \in \mathbb{R}^{N\times N}$, and $\|C\|$ is $o(\Delta)$.[6] Then for an initial transform $T_0$, there exists a sequence of quantization step sizes $\{\Delta_n\} \subset \mathbb{R}^+$ such that the iteration converges to a decorrelating transform,* i.e. $|||R_{y_n}||| \to 0$.

The proof, which is omitted due to its length (see [7]), is based on the existence at iteration $n$ of $\Delta_n \in \mathbb{R}^+$ such that $|||R_{y_{n+1}}||| \leq \frac{1}{2}|||R_{y_n}|||$. Thus the proof does not preclude the possibility that we must have $\lim_{n\to\infty} \Delta_n = 0$ for convergence. However,

[5] "Estimation noise" denotes the inaccuracy in the estimation of moments due to having only a finite sample.

[6] Here $\|\cdot\|$ is used to denote the Frobenius norm $\|A\|^2 = \sum_{i,j} a_{ij}^2$.

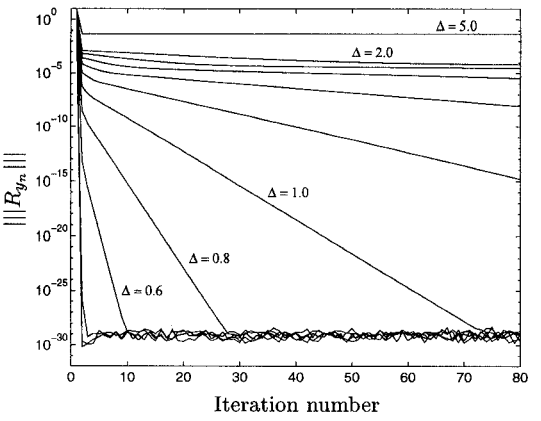


Figure 2: Simulations for various quantization step sizes demonstrating the convergence predicted by Theorem 2. The source is a first-order Gauss-Markov process with correlation $\rho = 0.9$, the vector length is $N = 4$, and the initial transform is the identity transform.

we conjecture that the iteration will converge as above for a sufficiently small (but strictly positive) *fixed* quantization step size $\Delta$. This stronger convergence result is shown locally for the vector length $N = 2$ case in the following subsection.

The conditions of Theorem 2 are met, for example, when the source is Gaussian. Figure 2 shows simulation results for a first-order Gauss-Markov source with correlation coefficient $\rho = 0.9$, a vector length of $N = 4$, and $T_0 = I_4$. In the simulation, $|||R_{y_n}|||$ converges monotonically to within the unit round of zero for fixed quantization step sizes up to $\Delta = 2$.[7] The results are inconclusive for $\Delta = 5$.

## 4.2 Analysis of the $N = 2$ case

When the vector length is $N = 2$, a simple analysis of the iteration is possible because the transform matrices can be parameterized by a single variable. In particular, beyond giving an alternate proof of Theorem 2 for the $N = 2$ case, we can show that if the initial transform is sufficiently close to the optimal transform the iteration converges for a fixed, sufficiently small quantization step size $\Delta$.

Let $R_x = \begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho \\ \sigma_1\sigma_2\rho & \sigma_2^2 \end{bmatrix}$. Without loss of generality, we can assume that the transform iterates are all in $SO_2(\mathbb{R})$, parameterized as $T_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$. We assume $\sigma_1^2 \neq \sigma_2^2$; if not, the situation is uninteresting because $R_x$ is diagonalized by $T_{\pi/4}$ independent of $\rho$. Let $W(\theta)$ equal the off-diagonal element of $T_\theta R_x T_\theta^T$. It is easy

[7] Convergence for $\Delta = 2$ is very slow and is thus not apparent.

to verify that $\theta^\star = \frac{1}{2}\arctan(\frac{-2\sigma_1\sigma_2\rho}{\sigma_1^2-\sigma_2^2})$ is a zero of $W$ and hence $T_{\theta^\star}$ diagonalizes $R_x$. (We will not concern ourselves with the ordering of the diagonal elements of $R_y$.) We can furthermore show that $W(\theta) = c\sin(2(\theta-\theta^\star))$, where $c \equiv [\frac{1}{4}(\sigma_1^2-\sigma_2^2)^2 + \sigma_1^2\sigma_2^2\rho^2]^{1/2}$.

We will show that the iteration converges in a neighborhood of $\theta^\star$ for small $\Delta$. Consider an iteration starting at $\theta$ (near $\theta^\star$). Let $\widetilde{W}(\theta) = W(\tilde{\theta})$, where $\tilde{\theta}$ is the iterate that follows from $\theta$. Then $\widetilde{W}(\theta) = c\sin\left(\arctan\left(\frac{-2\hat{X}_{12}}{\hat{X}_{11}-\hat{X}_{22}}\right) - 2\theta^\star\right)$, where $R_{\hat{x}} = \begin{bmatrix} \hat{X}_{11} & \hat{X}_{12} \\ \hat{X}_{21} & \hat{X}_{22} \end{bmatrix}$. Differentiating with respect to $\theta$ one can show that

$$\left|\frac{d}{d\theta}\widetilde{W}(\theta)\right| \leq 2c\left|\frac{d}{d\theta}\left[\frac{\hat{X}_{12}}{\hat{X}_{11}-\hat{X}_{22}}\right]\right|. \tag{2}$$

Defining $\delta \in \mathbb{R}^+$, $\alpha, \beta, \gamma \in \mathbb{R}$ through[8] $\begin{bmatrix} \delta+\alpha & \beta \\ \beta & \delta+\gamma \end{bmatrix} = R_{\hat{y}} - R_y$, one can verify that

$$\frac{\hat{X}_{12}}{\hat{X}_{11}-\hat{X}_{22}} = \frac{\sigma_1\sigma_2\rho - \frac{1}{2}(\alpha-\gamma)\sin 2\theta + \beta\cos 2\theta}{\sigma_1^2-\sigma_2^2+(\alpha-\gamma)\cos 2\theta + 2\beta\sin 2\theta}. \tag{3}$$

From (3) it is clear that by having $\alpha$, $\beta$, and $\gamma$ small enough but still potentially nonzero (fine enough quantization but with a strictly positive step size), one can insure that $\left|\frac{d}{d\theta}\frac{\hat{X}_{12}}{\hat{X}_{11}-\hat{X}_{22}}\right| < 1$, independent of $\theta$. Substituting in (2) and comparing to $\frac{d}{d\theta}W(\theta)$, we get $\left|\frac{d}{d\theta}\widetilde{W}(\theta)\Big|_{\theta=\theta^\star}\right| < \left|\frac{d}{d\theta}W(\theta)\Big|_{\theta=\theta^\star}\right|$, so the iteration converges in a neighborhood of $\theta^\star$.

# 5 Computational Refinements

In the simulations reported in [1], conceptually simple data structures and computational mechanisms were used. In particular,

$$\hat{r}_x(m) = \frac{1}{K}\sum_{k=0}^{K-m-1} x(k)x(k+m), \quad m = 0, 1, \ldots, N-1, \tag{4}$$

was used as an autocorrelation estimate computed from samples $\{x_k\}_{k=0}^{K-1}$, and eigendecompositions were computed using the MATLAB `eig` function. In this section a few possible computational refinements are presented. The computational savings and effects on performance are under investigation.

## 5.1 Autocorrelation estimation and data windowing

Suppose we are coding with vectors of length $N$, updating the transform after every $M$ samples, and using the past $L$ samples for each transform update. Except for the

[8] This form does not uniquely define $\delta$, $\alpha$, and $\gamma$. It was chosen because, in the Gaussian case, one can take $\delta = \frac{\Delta^2}{12}$ and be left with $\alpha$ and $\gamma$ which go to zero even faster as $\Delta \to 0$.

first update interval, where the two are equivalent, (4) can be replaced by a slightly better estimate

$$\hat{r}_{x,kM}(m) = \frac{1}{L}\sum_{\ell=0}^{L-1} x(kM - L + \ell)x(kM - L + \ell - m), \quad m = 0, 1, \ldots, N-1, \quad (5)$$

where the added subscript $kM$ denotes that the estimate is computed at time $kM$, *i.e.* with data up to and including $x(kM-1)$. It is understood that $x(-(N-1))$, $x(-(N-2))$, ..., $x(-1)$ are to be taken to be zero.

Since the update of the transform depends only on estimated moments of the source signal, the coder and decoder need not actually store the $L$ past samples. Neglecting block boundary effects, computing (5) simply requires $N$ accumulators, one for each desired autocorrelation lag, and the past $N-1$ samples with which to form products.

If $L = M$, handling block boundaries is very simple. After each transform update the accumulators can all simply be reset to zero. Since the $N-1$ previous samples are being buffered, calculations can proceed as before. The $L < M$ case is even simpler. After each transform update, one need not even increment the accumulators for the next $M - L$ samples. The accumulators are then reset and calculations proceed as before.

The memory requirements can not be reduced as dramatically when $L > M$. For simplicity, suppose $L = PM$ with $P \in \mathbb{Z}^+$.[9] In this case, the sum in (5) can be broken into $P$ sums, giving

$$\hat{r}_{x,kM}(m) = \frac{1}{P}\sum_{K=0}^{P-1} r'_{x,(k-P+1+K)M}(m), \quad m = 0, 1, \ldots, N-1, \quad (6)$$

where

$$\hat{r}'_{x,kM}(m) = \frac{1}{M}\sum_{\ell=0}^{M-1} x((k+P-1)M - L + \ell)x((k+P-1)M - L + \ell - m).$$

(The notation is somewhat cumbersome, but essentially $\hat{r}_{x,t}$ and $\hat{r}'_{x,t}$ are autocorrelation estimates using the last $L$ and $M$ data samples, respectively, up to but not including time $t$.) This shows that running autocorrelation estimates can be calculated using $P$ sets of $N$ accumulators and an $N-1$ sample buffer. This is still a great savings over storing $L$ samples.

Equation (6) is an average of $P$ autocorrelation estimates, each based on a window of length $M$ samples (plus a few earlier samples, actually). There is no *a priori* requirement that each of the $P$ estimates be weighted equally. One way to more heavily weight the later autocorrelation estimates is to use a "forgetting factor" $\alpha$ as in Recursive Least Squares [8]. The estimate would then be updated through

$$\hat{r}_{x,\text{new}} = (1-\alpha)\hat{r}_{x,\text{old}} + \alpha\hat{r}'_x$$

as each new $\hat{r}'_x$ is calculated. This scheme eliminates the need for extra accumulators, again lowering the requirement to $N$, one for each autocorrelation lag.

[9] If $L/M \notin \mathbb{Z}^+$, similar arguments could be made with $P = \gcd(L, M)$.

### 5.2 Efficient update mechanisms

If the variation of the source is slow relative to the update interval—and the update interval is long enough to have reasonable noise suppression—successive autocorrelation estimates will not differ greatly. This feature can be exploited in the calculation of the KLT. For example, one can first approximately diagonalize the autocorrelation matrix using the previous KLT estimate. Then it should be possible to complete the diagonalization with a small number of Jacobi rotation steps (see [9, §8.5]).

## 6 Conclusions and Ongoing Work

A method for universal transform coding which updates the transform based on a local KLT estimate was developed. The KLT estimate is derived purely from quantized data, so the decoder can track the encoder state without any side information.

The convergence of this method in the absence of estimation noise was analyzed and two convergence results were established. In simulations, the iteration converges even when the quantization is relatively coarse.

Work on applying this coding scheme for still images is ongoing. On the theoretical side, it seems that for stationary or slowly varying Gaussian sources it is possible to precisely characterize the relationship between the update interval, the ability to track the source, and the adaptation loss.

## References

[1] V. K Goyal, J. Zhuang, M. Vetterli, and C. Chan. Transform coding using adaptive bases and quantization. In *IEEE Int. Conf. Image Proc.*, vol. II, pp. 365–368, 1996.

[2] M. Effros and P. A. Chou. Weighted universal transform coding: Universal image compression with the Karhunen-Loève transform. In *IEEE Int. Conf. Image Proc.*, vol. II, pp. 61–64, 1995.

[3] R. D. Dony and S. Haykin. Optimally adaptive transform coding. *IEEE Trans. Image Proc.*, 4(10):1358–1370, October 1995.

[4] H. S. Malvar. *Signal Processing with Lapped Transforms.* Artech House, 1992.

[5] B. Yu. A statistical analysis of adaptive scalar quantization based on quantized past. Submitted to *IEEE Trans. Info. Theory*, 1995.

[6] L. Cheded and P. A. Payne. The exact impact of amplitude quantization on multi-dimensional, high-order moments estimation. *Sig. Proc.*, 39(3):293–315, 1994.

[7] J. Zhuang. Adaptive transforms and quantization. MS thesis, UC-Berkeley, 1997.

[8] C. R. Johnson, Jr. *Lectures on Adaptive Parameter Estimation.* Prentice Hall, 1988.

[9] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins Univ. Press, second edition, 1989.