A Generalized Lloyd-Type Algorithm for Adaptive Transform Coder Design

Cynthia Archer and Todd K. Leen

***Abstract*—In this paper, we establish a probabilistic framework for adaptive transform coding that leads to a generalized Lloyd type algorithm for transform coder design. Transform coders are often constructed by concatenating an *ad hoc* choice of transform with suboptimal bit allocation and quantizer design. Instead, we start from a probabilistic latent variable model in the form of a mixture of constrained Gaussian mixtures. From this model, we derive an transform coder design algorithm, which integrates optimization of all transform coder parameters. An essential part this algorithm is our introduction of a new transform basis—the coding optimal transform—which, unlike commonly used transforms, minimizes compression distortion.**

Adaptive transform coders can be effective for compressing databases of related imagery since the high overhead associated with these coders can be amortized over the entire database. For this work, we performed compression experiments on a database of synthetic aperture radar images. Our results show that adaptive coders improve compressed signal-to-noise ratio (SNR) by approximately 0.5 dB compared with global coders. Coders that incorporated the coding optimal transform had the best SNRs on the images used to develop the coder. However, coders that incorporated the discrete cosine transform generalized better to new images.

***Index Terms*—Adaptive transform coding, compression, entropy-constrained quantization, expectation-maximization, Gaussian mixtures, generalized Lloyd algorithms.**

I. Introduction

TRANSFORM coding is a computationally attractive alternative to vector quantization and is widely used for image and video compression. A transform coder compresses multidimensional data by first transforming the data vectors to new coordinates and then coding the transform coefficient values independently with scalar quantizers. A key goal of the transform coder is to minimize compression distortion while keeping the compressed signal representation below some target size. In this paper, we quantify compression distortion as the mean squared error due to quantization. While quantizers have typically been designed to minimize compression distortion [1], [2], this has not been the case for the transform portion of the coder. The transform has either been fixed *a priori*, as in the discrete cosine transform (DCT) used in the JPEG compression standard [3], or adapted to the signal statistics using the Karhunen–Loéve transform (KLT), as in recently published transform coding work [4], [5]. These transforms are not designed to minimize compression distortion, nor are they designed (selected) in concert with quantizer development to deliver the best compression performance.

Classic transform design assumes that correlations between signal components are the same everywhere in the signal space. This assumption is valid only when the data is wide sense stationary. Noting that signals such as images and speech are nonstationary, several researchers have extended global transform coding to adapt to changing signal characteristics [4]–[7]. In *adaptive* transform coding, the signal space is partitioned into disjoint regions, and a set of basis functions (transforms) and scalar quantizers are designed for each region. In our own previous work [7], we use k-means clustering [8] to define these regions. Dony and Haykin [4] partition the space to minimize dimension-reduction error. Tipping and Bishop [6] use soft partitioning according to a probabilistic rule that reduces, in the appropriate limit, to partitioning by dimension-reduction error, as defined by Khambatla and Leen in [9]. These last two techniques minimize dimension reduction error rather than compression distortion. Effros *et al.* [5] partition the signal space to minimize entropy-constrained compression distortion but then use heuristics to design the local transform coders. Since the coders are not designed to minimize the compression distortion, there is no guarantee that the algorithm will converge to a distortion minimum. None of these systems integrate optimization of all the transform coder parameters nor design those parameters to produce a coder that minimizes compression distortion.

In contrast to the piecemeal construction of transform coders, vector quantizers (VQs) are designed with algorithms [10], [11] that minimize compression distortion. Nowlan [12] uses a probabilistic framework to derive a VQ design algorithm from a mixture of Gaussians model of data. In the limit that the variance of the mixture components goes to zero, the expectation–maximization (EM) procedure [13] for fitting the mixture model to data reduces to the K-means algorithm [8] or, equivalently, the Linde–Buzo–Gray (LBG) algorithm [10] for vector quantizer design. In addition, Chou *et al.* [11] note that the design algorithm for an entropy-constrained VQ (ECVQ) is a hard-clustering version of the EM algorithm for fitting a mixture of spherical Gaussians with nonzero component variance to data. In this latter case, the component variance acts as the Lagrange multiplier linking the mean squared error and entropy constraint terms. Consequently, choosing the component variance corresponds to selecting the entropy constraint or compressed bit-rate.

We make use of this probabilistic framework to cast transform coding as a constrained form of vector quantization. We

Manuscript received July 1, 2002; revised March 12, 2003. This work was supported in part by National Science Foundation under Grant ECS-9976452. The associate editor coordinating the review of this paper and approving it for publication was Prof. Sheila S. Hemami.

C. Archer is with the Research Triangle Institute International, Lake Oswego, OR 97035 USA (e-mail: carcher@rti.org).

T. K. Leen is with the Department of Computer Science and Engineering, OGI School of Science and Engineering, Oregon Health and Science University, Beaverton, OR 97006 USA (e-mail: tleen@cse.ogi.edu).

Digital Object Identifier 10.1109/TSP.2003.819980

first define a *constrained* mixture of Gaussians model based on the VQ probability model. This model provides a framework for developing a new generalized Lloyd algorithm for transform coder design. This algorithm integrates optimization of the signal space partition (encoder) and the local transforms and scalar quantizers (decoder). A significant contribution of our work is a new orthogonal transform that minimizes compression distortion rather than dimension reduction error. We conclude by validating our algorithms by compressing a database of synthetic aperture radar (SAR) images.

II. Probability Models for Transform Coding

An effective paradigm for designing new algorithms involves defining a statistical model of the signal behavior and using a maximum likelihood framework to guide algorithm development. We take this approach and develop constrained mixture of Gaussians models that provide a statistical model for both global and adaptive transform coding. While it is possible to develop a generalized Lloyd type algorithm for transform coding without this model, starting from a statistical model has several advantages. Developing a probability model makes explicit our assumptions about data behavior and characteristics. In addition, the probability model indicates an appropriate distortion metric. For instance, the likelihood of a Gaussian model incorporates a mean squared error metric (as shown below), whereas a magnitude of error metric is consistent with Laplacian models. Comparing observed data behavior to the model can pinpoint under what conditions the model poorly describes the data, which can guide improvements in the related algorithm. Finally, the statistical model can provide a framework for comparing and understanding the relationship between different algorithms [14].

This section begins with a brief review of transform coder operation. We then present a probablity model for global transform coding that is a constrained form of the VQ model described by Nowlan [12] and Chou *et al.* [11]. This is followed by our development of a probability model for adaptive tranform coding. The transform coder design algorithm derived from this model is described in the following section.

A. Global Transform Coding

The compression and restoration processes replace each signal vector x with one of a small set of reproduction vectors $\mathbf{q}_\alpha$, $\alpha = 1 \ldots \mathcal{K}$. A reproduction vector $\mathbf{q}_\alpha$ represents all the data vectors in some region R_α of the data space. To compress a signal by assigning data to reproduction vectors, a transform coder converts the d-dimensional data to new coordinates and then codes the transform coefficients independently of one another with *scalar* quantizers. One can think of a transform coder as a vector quantizer with the $\mathcal{K}$ reproduction vectors *constrained to lie at the vertices of a rectangular grid*. The grid is defined by orthogonal axes s_J, $J = 1 \ldots d$ and d sets of scalar reproduction values: one set for each dimension. There are $\mathcal{K}_J$ distinct reproduction values on the s_J axis; thus, the total number of grid vertices or reproduction vectors is $\mathcal{K} = \prod_J \mathcal{K}_J$.

Fig. 1 illustrates the structure of a two-dimensional (2-D) transform coder. The r values indicate the scalar reproduction values; r_{Ji} is the ith value along the s_J axis. (We use capital Roman letters to indicate coordinate axes and lowercase Roman letters to indicate reproduction values along those axes.) The coordinates of the reproduction *vectors* $\mathbf{q}_\alpha$ are combinations of the scalar reproduction values $[r_{1i}, r_{2j}, \ldots, r_{dk}]^T, i = 1 \ldots \mathcal{K}_1$, $j = 1 \ldots \mathcal{K}_2$, etc. The $d \times d$ orthogonal transform W defines the orientation of this quantizer grid in the data space. In the data coordinate basis, the reproduction vectors are given by $W\mathbf{q}_\alpha$.

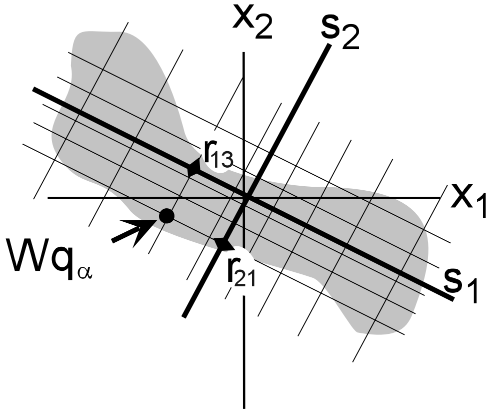


Fig. 1. Orientation of quantizer grid in signal space. The quantizer reproduction vectors $\mathbf{q}_\alpha$, $\alpha = 1 \ldots \mathcal{K}$ lie at the vertices of a rectangular grid. The grid is oriented to the signal vectors x (indicated by the gray area) with orthogonal transform, W.

Since transform coding is a *constrained* form of vector quantization, it will introduce more distortion than an unconstrained VQ at a given compressed bit-rate. However, the coding complexity is substantially less. Encoding a d-dimensional data vector with a vector quantizer requires $\mathcal{O}(\mathcal{K}d)$ add/multiply operations for the distance calculations and $\mathcal{O}(\mathcal{K})$ compare operations. A transform coder requires $\mathcal{O}(d^2)$ add/multiply operations for the transform, where d is normally much smaller than $\mathcal{K}$. Coding the scalar transform coefficients requires $\mathcal{O}(\log_2 \mathcal{K})$ compare operations.

B. Global Transform Coder Model

To replicate the transform coder structure, we envision the data as drawn from a d-dimensional latent data space S and embedded in an observation or measurement space X, which is also d dimensional. The density on the latent space is a mixture of delta functions

$$\mathrm{p}(s) = \sum_{\alpha=1}^{\mathcal{K}} \pi_\alpha \delta(s - \mathbf{q}_\alpha) \tag{1}$$

where the latent values $\mathbf{q}_\alpha$ lie at the vertices of a rectangular grid, as illustrated in Fig. 2. The grid is defined by the s axes and a set of grid mark values $\{r_{Ji}\}$, where r_{Ji} is the ith grid mark along the s_J axis. There are $\mathcal{K}_J$ distinct grid mark values on the s_J axis, making the total number of grid vertices $\mathcal{K} = \prod_J \mathcal{K}_J$. Thus, the coordinates of some $\mathbf{q}_\alpha$ can be written as $[r_{1i}, r_{2j}, \ldots, r_{dk}]^T$. By incorporating this constraint into (1), we can write the density on s as product of marginal densities

$$\mathrm{p}(s) = \prod_{J=1}^{d} \sum_{i=1}^{\mathcal{K}_J} \mathrm{p}_{Ji} \delta(s_J - r_{Ji}) \tag{2}$$

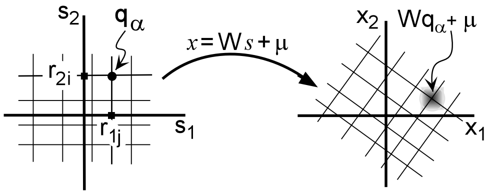


Fig. 2. Structure of latent variable space S and mapping to observed space X. The data density in the latent space consists of a mixture of delta functions where the mixture components $\mathbf{q}_\alpha$ are constrained to lie at the vertices of a rectangular grid. This grid is mapped to the observed data space by an orthogonal transform W and corrupted with additive Gaussian noise.

where the mixing coefficients π_α are the product of prior probabilities p_{Ji}

$$\pi_\alpha = \prod_J p_{Ji}. \tag{3}$$

We will use both latent density formulations (1) and (2) for algorithm development.

The latent data is mapped to the observation space by an orthogonal transformation W, as illustrated in Fig. 2. The embedded data is corrupted with additive Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2\mathrm{I})$, with mean zero and variance $\sigma^2\mathrm{I}$. We use the same noise distribution as the VQ model [12], since this choice leads to a nonweighted mean squared error (Euclidean distance) metric, as shown below.

The observed data generated from a sample s drawn from latent component α is

$$x = W(s - \mathbf{q}_\alpha) + \mu + \epsilon \tag{4}$$

with conditional densities

$$\mathrm{p}(x|s,\alpha) = \mathcal{N}\left(\mu + W(s - \mathbf{q}_\alpha), \sigma^2\mathrm{I}\right). \tag{5}$$

The latent density and mapping induces a mixture of constrained Gaussian density on x of the form

$$\begin{aligned} \mathrm{p}(x) &= \int \sum_\alpha \pi_\alpha \mathrm{p}(x|s,\alpha)\delta(s - \mathbf{q}_\alpha)ds \\ &= \sum_{\alpha=1}^{\mathcal{K}} \pi_\alpha \mathrm{p}(x|\alpha) \end{aligned} \tag{6}$$

with marginal density

$$p(x|\alpha) = \mathcal{N}(\mu + W\mathbf{q}_\alpha, \sigma^2\mathrm{I}). \tag{7}$$

The expectation–maximization algorithm (EM) [13] fits parametric probability models to data by maximizing the log likelihood of the model for some training data set $\{x_n, n = 1 \ldots N\}$. The log likelihood is given by

$$\mathcal{L} = \sum_{n=1}^{N} \log\left(\sum_{\alpha=1}^{\mathcal{K}} \pi_\alpha \mathrm{p}(x_n|\alpha)\right) \tag{8}$$

where log is the natural logarithm. In order to simplify (8), we introduce the density $z(\alpha, x_n)$ over the unknown component assignments so that

$$\mathcal{L} = \sum_{n=1}^{N} \log\left(\sum_{\alpha=1}^{\mathcal{K}} z(\alpha, x_n)\frac{\pi_\alpha \mathrm{p}(x_n|\alpha)}{z(\alpha, x_n)}\right) \tag{9}$$

where $\sum_\alpha z(\alpha, x) = 1$. Using Jensen's inequality to bring the sum over α outside the logarithm function, we find $\mathcal{L}$ is bounded below by the *expected* log likelihood

$$\begin{aligned} \mathcal{L} \geq \langle\mathcal{L}\rangle = &\sum_{n=1}^{N}\sum_{\alpha=1}^{\mathcal{K}} z(\alpha, x_n) \\ &\times\left[\log\pi_\alpha - \frac{d}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|x_n - \mu - W\mathbf{q}_\alpha\|^2\right] \\ &- \sum_{n=1}^{N}\sum_{\alpha=1}^{\mathcal{K}} z(\alpha, x_n)\log z(\alpha, x_n) \end{aligned} \tag{10}$$

with equality when $z(\alpha, x) = p(\alpha|x)$ is the posterior probability of component α conditioned on the data vector x [15]. The norm $\|x_n - \mu - W\mathbf{q}_\alpha\|^2$ is given by the inner product $(x_n - \mu - W\mathbf{q}_\alpha)^T(x_n - \mu - W\mathbf{q}_\alpha)$.

The EM algorithm provides a template for deriving the transform coding algorithm from this probability model. To achieve the hard-clustering needed for transform coding, we choose $z(\alpha, x_n)$ to be

$$z(\alpha, x_n) = \begin{cases} 1, & \mathrm{p}(\alpha|x_n) > \mathrm{p}(\gamma|x_n)\ \forall\gamma \neq \alpha \\ 0, & \text{otherwise} \end{cases}. \tag{11}$$

With this hard-clustering model, the final term in the expected log likelihood (10) becomes zero since $z(\alpha, x_n)\ln z(\alpha, x_n) = 0 \forall\alpha, n$. If we remove unessential terms and scale by $2\sigma^2/N$, $\langle\mathcal{L}\rangle$ reduces to the cost function

$$\mathcal{C} = \frac{1}{N}\sum_{\alpha=1}^{\mathcal{K}}\sum_{n=1}^{N} z(\alpha, x_n)\left(\|x_n - \mu - W\mathbf{q}_\alpha\|^2 - 2\sigma^2\log\pi_\alpha\right). \tag{12}$$

This cost function consists of two terms: the average coding distortion as measured by mean squared error

$$\mathcal{D} = \frac{1}{N}\sum_\alpha\sum_n z(\alpha, x_n)\|x_n - \mu - W\mathbf{q}_\alpha\|^2 \tag{13}$$

and the entropy

$$\mathcal{H} = -\sum_\alpha \pi_\alpha \log\pi_\alpha. \tag{14}$$

This *entropy-constrained* cost function (12) is the same as that found by minimizing mean squared error subject to an *average* bit-rate constraint (e.g., [11]). Note that both the mean squared error metric and the entropy constraint arise directly from the probability model. The noise variance σ^2 acts as a Lagrange multiplier linking the distortion and entropy terms. When the noise variance is chosen to be large, the entropy term has a large effect on the cost resulting in a high-distortion, low-rate coder. Conversely, when the noise variance is small, the distortion term dominates the cost function resulting in a low-distortion, high-rate coder.

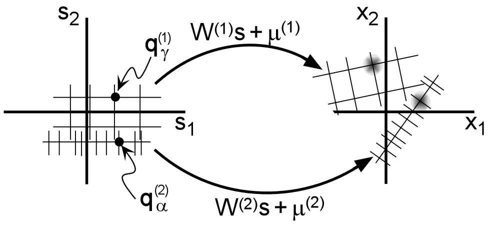


Fig. 3. Nonstationary data model. Structure of latent variable space S and mapping to observed space X. The mixture components $\mathbf{q}_\alpha^{(\mathrm{m})}$ are constrained to lie at the vertices of the Mth grid. Latent data is mapped to the observation space by orthogonal transforms $W^{(\mathrm{m})}$ and corrupted with additive Gaussian noise.

C. Adaptive Transform Coder Model

An *adaptive* transform coder consists of a collection of transform coders, each specialized to optimally compress data from a different region of the data space. Consequently, the model for adaptive transform coding is a collection or mixture of global transform coding models. The d-dimensional latent data s lies at the vertices $\mathbf{q}_\alpha^{(\mathrm{m})}$ of one of M rectangular grids centered at $\eta^{(\mathrm{m})}$. There are $\mathcal{K}_J^{(\mathrm{m})}$ distinct grid mark values along the s_J axis in the mth grid making the total number of grid vertices $\mathcal{K}_\mathrm{m} = \prod_J \mathcal{K}_J^{(\mathrm{m})}$. Each grid can have a different number of components $\mathcal{K}_\mathrm{m}$. The total number of reproduction values is $\mathcal{K} = \sum_\mathrm{m} \mathcal{K}_\mathrm{m}$. The density on the whole latent space consists of a *mixture* of delta function mixtures

$$\mathrm{p}(s) = \sum_{\mathrm{m}=1}^{M} \pi_\mathrm{m} \sum_{\alpha=1}^{\mathcal{K}_\mathrm{m}} p(\alpha|\mathrm{m}) \delta\left(s - \eta^{(\mathrm{m})} - \mathbf{q}_\alpha^{(\mathrm{m})}\right) \tag{15}$$

where π_m are the grid mixing coefficients.

The latent data from each grid m is mapped to the observation space by its own orthogonal transform $W^{(\mathrm{m})}$. As in the global model, the data is then corrupted with additive Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$. The observed data generated from some sample s drawn from latent component (α, m) is $x = W^{(\mathrm{m})}(s - \eta^{(\mathrm{m})} - \mathbf{q}_\alpha^{(\mathrm{m})}) + \mu^{(\mathrm{m})} + \epsilon^{(\mathrm{m})}$.

Fig. 3 illustrates this mapping from a two-grid latent space. The latent density and mapping induce a mixture of constrained Gaussian mixtures density on x of the form

$$\mathrm{p}(x) = \sum_{\mathrm{m}} \pi_\mathrm{m} \sum_{\alpha=1}^{\mathcal{K}_\mathrm{m}} \mathrm{p}(\alpha|\mathrm{m}) \mathrm{p}(x|\alpha, \mathrm{m}) \tag{16}$$

with the marginal densities

$$p(x|\alpha, \mathrm{m}) = \mathcal{N}\left(\mu^{(\mathrm{m})} + W^{(\mathrm{m})}\mathbf{q}_\alpha^{(\mathrm{m})}, \sigma^2 \mathrm{I}\right). \tag{17}$$

The log likelihood of some training data set $\{x_n, n = 1 \ldots N\}$ is given by

$$\mathcal{L} = \sum_{n=1}^{N} \log\left(\sum_{\mathrm{m}=1}^{M} \pi_\mathrm{m} \sum_{\alpha=1}^{\mathcal{K}_\mathrm{m}} \mathrm{p}(\alpha|\mathrm{m}) \mathrm{p}(x_n|\alpha, \mathrm{m})\right). \tag{18}$$

We simply as before in order to achieve the hard-clustering needed for transform coding. Consequently, we choose $z(\alpha, \mathrm{m}, x_n)$ to be one or zero

$$z(\alpha, \mathrm{m}, x_n) = \begin{cases} 1, & \mathrm{p}(\alpha|\mathrm{m}x_n) > \mathrm{p}(\gamma, \hat{\mathrm{m}}|\mathrm{m}) \\ & \quad \forall \gamma \neq \alpha \text{ and } \hat{\mathrm{m}} \neq \mathrm{m} \\ 0, & \quad \text{otherwise} \end{cases}. \tag{19}$$

By removing unnecessary terms and scaling by $2\sigma^2/N$, $\langle \mathcal{L} \rangle$ reduces to the cost function

$$\begin{aligned} \mathcal{C} = {} & \frac{1}{N} \sum_{\mathrm{m}=1}^{M} \sum_{\alpha=1}^{\mathcal{K}_\mathrm{m}} \sum_{n=1}^{N} z(\alpha, \mathrm{m}, x_n) \left\| x_n - \mu^{(\mathrm{m})} - W^{(\mathrm{m})}\mathbf{q}_\alpha^{(\mathrm{m})} \right\|^2 \\ & - \frac{1}{N} \sum_{\mathrm{m}=1}^{M} \sum_{\alpha=1}^{\mathcal{K}_\mathrm{m}} \sum_{n=1}^{N} z(\alpha, \mathrm{m}, x_n) 2\sigma^2 \log\left[\pi_\mathrm{m} \mathrm{p}(\alpha|\mathrm{m})\right]. \end{aligned} \tag{20}$$

This cost function consists of two terms: the mean squared error

$$\mathcal{D} = \frac{1}{N} \sum_{\mathrm{m}} \sum_{\alpha} \sum_{n} z(\alpha, \mathrm{m}, x_n) \left\| x_n - \mu^{(\mathrm{m})} - W^{(\mathrm{m})}\mathbf{q}_\alpha^{(\mathrm{m})} \right\|^2 \tag{21}$$

and the discrete entropy

$$\mathcal{H} = -\sum_{\mathrm{m}} \sum_{\alpha} \left(\pi_\mathrm{m} \mathrm{p}(\alpha|\mathrm{m})\right) \log\left(\pi_\mathrm{m} \mathrm{p}(\alpha|\mathrm{m})\right). \tag{22}$$

The entropy term includes both the bit-rate required to code the transform coefficients $(-\pi_\mathrm{m} \mathrm{p}(\alpha|\mathrm{m}) \log \mathrm{p}(\alpha|\mathrm{m}))$ and the bit-rate required to indicate the choice of local transform coder $(-\pi_\mathrm{m} \log \pi_\mathrm{m})$. Note that the entropy does not include the overhead cost of encoding the parameter values. As in the global transform coding case, the noise variance acts as a Lagrange multiplier linking the distortion and entropy terms. When the noise variance is chosen to be large, the entropy term has a large effect on the cost resulting in a high-distortion, low-rate coder. Conversely, when the noise variance is small, the distortion term dominates the cost function resulting in a low-distortion, high-rate coder. In the limit that the noise variance σ^2 goes to zero, *and we limit the number of code vectors*, we recover the cost function for *fixed-rate* adaptive transform coding [16]. When the number of grids $M = 1$, we recover the cost function for *global* transform coding.

III. Adaptive Transform Coding Algorithm

In this section, we present a new algorithm for adaptive transform coder design that integrates optimization of the transform coder parameters: the data space partition, transforms, and quantizers. This generalized Lloyd type algorithm fits the parameters to data so that entropy-constrained coding distortion (20) is minimized. Like all such algorithms, the optimization process is iterative. It alternately partitions the data space into local regions and then optimizes the transform and quantizers for each region. Each such iteration reduces (or at least does not increase) the value of the cost function. Generalized Lloyd type algorithms converge to a local minimum of the cost function. Note that we recover the generalized Lloyd algorithm for *global* transform coding when the number of local coders is set to one.

Partition Optimization: To optimize the partition, each data vector is assigned to the reproduction vector $q_\alpha^{(\mathrm{m})}$ of transform coder m that represents it with the least entropy-constrained distortion. To partition the data, we compress each d-dimensional data vector x with each local transform coder $\mathrm{m} = 1 \ldots M$. To compress x, we first find the transform coefficients, $s_J^{(\mathrm{m})} = (W_J^{(\mathrm{m})})^T x$, $J = 1 \ldots d$, where W_J is the Jth basis (column) vector of the W transform matrix. Each $s_J^{(\mathrm{m})}$ is then assigned to the scalar quantizer reproduction value $r_{Ji}^{(\mathrm{m})}$ that represents it with the least entropy-constrained distortion. Fig. 4 demonstrates this transform and coding process.

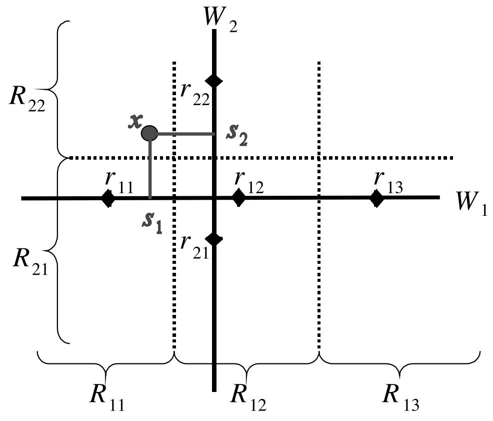


Fig. 4. Transform coding a data vector. Projecting data vector x with transform W yields coefficient values $W_1^T x = s_1$ and $W_2^T x = s_2$. The data space is partitioned into subregions with boundaries indicated by dotted lines. Coefficient s_1 is in subregion R_{11}, and s_2 is in subregion R_{22}; hence, x is represented by reproduction vector $[r_{11}, r_{22}]^T$.

The cost of assigning x to transform coder m is

$$C^{(\mathrm{m})}(x) = \sum_{J=1}^{d} \left(\left\| \left(W_J^{(\mathrm{m})} \right)^T x - r_{Ji}^{(\mathrm{m})} \right\|^2 + 2\sigma^2 l_{Ji}^{(\mathrm{m})} \right) \quad (23)$$

where $l_{Ji}^{(\mathrm{m})} = -\log \mathrm{p}(r_{Ji}^{(\mathrm{m})}|\mathrm{m})$. We then assign x to transform coder $\hat{\mathrm{m}}$ such that

$$\hat{\mathrm{m}} = \arg\min_{m} C^{(\mathrm{m})}(x) - 2\sigma^2 \log \pi_{\mathrm{m}}. \quad (24)$$

Hence, the data space partition defines regions $R^{(\mathrm{m})}$ such that each x belongs to the transform coder that compresses it with the least entropy-constrained distortion

$$R^{(\mathrm{m})} = \Big\{ x | \left(C^{(\mathrm{m})}(x) - 2\sigma^2 \log \pi_{\mathrm{m}} \right) < \left(C^{(\hat{\mathrm{m}})}(x) - 2\sigma^2 \log \pi_{\hat{\mathrm{m}}} \right) \ \forall \hat{\mathrm{m}} \neq \mathrm{m} \Big\}. \quad (25)$$

In addition, the partition defines subregions $R_{Ji}^{(\mathrm{m})}$ such that each local transform coefficient $s_J^{(\mathrm{m})} = (W_J^{(\mathrm{m})})^T x$, $x \in R^{(\mathrm{m})}$ belongs to the scalar reproduction value $r_{Ji}^{(\mathrm{m})}$ that represents it with the lowest entropy-constrained distortion

$$R_{Ji}^{(\mathrm{m})} = \Big\{ s_J^{(\mathrm{m})} | \left(\left\| s_J^{(\mathrm{m})} - r_{Ji}^{(\mathrm{m})} \right\|^2 + 2\sigma^2 l_{Ji}^{(\mathrm{m})} \right) < \left(\left\| s_J^{(\mathrm{m})} - r_{Jk}^{(\mathrm{m})} \right\|^2 + 2\sigma^2 l_{Jk}^{(\mathrm{m})} \right) \ \forall k \neq i \Big\}. \quad (26)$$

Fig. 5 illustrates the relationship between the transform coder regions, $R^{(\mathrm{m})}$ and subregions $R_{Ji}^{(\mathrm{m})}$. Consequently, the new data space partition minimizes the cost function (20) for the current transform and quantizer values.

The prior probabilities $\mathrm{p}(r_{Ji}^{(\mathrm{m})}|\mathrm{m})$ and π_{m} are estimated from the number of data values in each region. The transform coder prior $\pi_{\mathrm{m}} = N_{\mathrm{m}}/N$, where N are the total number of data vectors, and N_{m} are the number of vectors in $R^{(\mathrm{m})}$. The reproduction value priors $\mathrm{p}(r_{Ji}^{(\mathrm{m})}|\mathrm{m}) = N_{Ji}^{(\mathrm{m})}/N_{\mathrm{m}}$, where $N_{Ji}^{(\mathrm{m})}$ are the number of transform coefficients in $R_{Ji}^{(\mathrm{m})}$.

Transform Optimization: To optimize the transform, we find the center μ and orientation W of each quantizer grid that minimizes the cost function (20). The minimum cost estimators for the grid center place each grid at the mean of the data assigned to each region

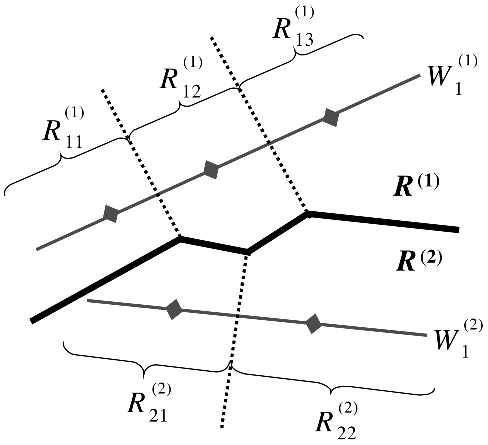


Fig. 5. Data space partition. Partition of a 2-D data space with two coders. Coder 1 consists of a 3 × 1 grid and coder 2 consists of a 2 × 1 grid. The boundary between the two coders, which partitions the data space into $R^{(1)}$ and $R^{(2)}$, is shown by the heavy black line. Subregion boundaries are indicated with dotted lines. The diamonds along the transform axes indicate placement of reproduction values.

$$\mu^{(\mathrm{m})} = \frac{1}{N_{\mathrm{m}}} \sum_{x \in R^{(\mathrm{m})}} x. \quad (27)$$

The minimum cost estimator for W is derived below; Appendix A contains a more detailed derivation. The grid orientation or transform W is constrained to be orthogonal, that is, $W^T W = \mathbf{I}$. Including the orthogonality constraint and dropping the entropy term, since it does not contain W, yields the cost function to be minimized

$$\mathcal{C}_{\mathrm{m}} = \frac{1}{N_{\mathrm{m}}} \sum_{\alpha=1}^{\mathcal{K}_{\mathrm{m}}} \sum_{x \in R_{\alpha}^{(\mathrm{m})}} \left\| x - \mu^{(\mathrm{m})} - \sum_{J=1}^{d} W_J^{(\mathrm{m})} q_{\alpha J}^{(\mathrm{m})} \right\|^2 + \sum_{K=1}^{d} \sum_{L=1}^{d} \gamma_{KL} \left(\left(W_K^{(\mathrm{m})} \right)^T W_L^{(\mathrm{m})} - \delta_{K,L} \right) \quad (28)$$

where W_J is the Jth column vector of W, $q_{\alpha K}$ is the Kth coordinate of reproduction vector q_α, and γ_{KL} is a Lagrange multiplier. If we define the outer-product matrix Q

$$Q = \sum_{\alpha} q_{\alpha}^{(\mathrm{m})} \sum_{x \in R_{\alpha}^{(\mathrm{m})}} (x - \mu^{(\mathrm{m})})^T \quad (29)$$

then minimizing the local cost function with respect to the transform yields

$$QW = W^T Q^T. \quad (30)$$

This symmetry condition (30) along with the orthogonality condition uniquely defines the coding optimal transform (COT).

To minimize distortion, the COT orients the quantizer grid so that the QW matrix is symmetric (30). We can quantify how far the matrix is from symmetric with the sum squared difference between transposed matrix elements

$$A = \sum_{K=1}^{d-1} \sum_{J=K+1}^{d} (a_{KJ} - a_{JK})^2 \quad (31)$$

where a_{KJ} is the Kth row and Jth column element of QW. We apply Givens rotations [17] $G(K, J, \theta)$ to minimize A. Multiplication by the $G(K, J, \theta)$ matrix applies a rotation of θ radians to the (K, J) coordinate plane. For a $d \times d$ matrix, there are $(d^2 - d)/2$ such planes. Minimizing (31) with respect to rotation $G(K, J, \theta)$ yields a solution for θ that is quartic in $\tan\theta$. However, when the angle is small $(\tan^2\theta \ll 1)$, the solution simplifies to

$$\tan\theta \approx \frac{(a_{KK}+a_{JJ})(a_{KJ}-a_{JK})}{\Phi} - \sum_{I\neq K,J} \frac{(a_{JI}a_{IK}-a_{JI}a_{IK})}{\Phi} \tag{32}$$

where $\Phi = \sum_{I\neq K,J}(a_{JI}a_{IJ} + a_{KI}a_{IK}) + (a_{KK} + a_{JJ})^2 - (a_{KJ} - a_{JK})^2$. Since the COT reduces to the KLT when the data is Gaussian [18], [19], we expect that starting the optimization from the KLT will keep the rotation angles small. This approach worked well in practice, allowing us to use this simpler form for the rotation angle. We find the rotation angle (32) for each coordinate plane and apply these rotations to the current transform matrix. This process is *repeated* until $A/\|QW\|_F$, where $\|QW\|_F$, which is the Frobenius norm, is less than a threshold $(A \approx 0)$. This new W will orient the quantizer grid so that compression distortion is minimized.

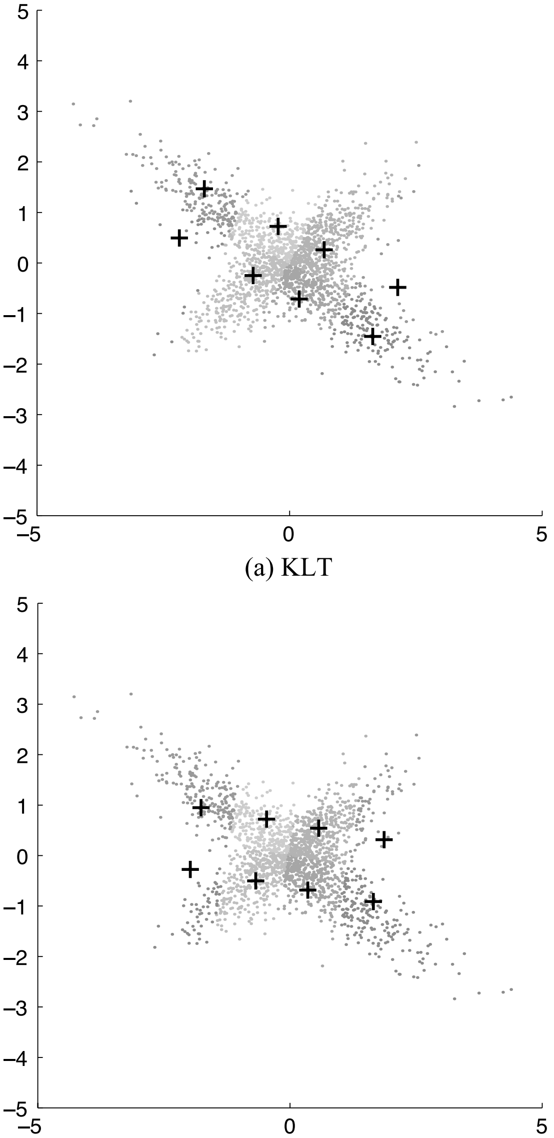
To illustrate the difference between the PCA transform and COT, we designed transform coders for 2-D data that is sampled from two intersecting Gaussian distributions: $\mathcal{N}(0, U_1^T\Sigma_1 U_1)$ with

$$U1 = \begin{bmatrix} -.6 & .8 \\ .8 & .6 \end{bmatrix} \text{ and } \Sigma_1 = \begin{bmatrix} 4 & 0 \\ 0 & .16 \end{bmatrix}$$

and $\mathcal{N}(0, U_2^T\Sigma_2 U_2)$ with

$$U_2 = \begin{bmatrix} .6 & .8 \\ .8 & -.6 \end{bmatrix} \text{ and } \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & .16 \end{bmatrix}.$$

Fig. 6 contains a plot of this data overlaid with a one by two bit quantizer grid. The KLT aligns the grid along the dominant high-variance Gaussian; consequently, data from the lower variance Gaussian is poorly represented. The COT rotates the quantizer grid so that the reproduction vectors better represent all the data. The compressed data signal-to-noise ratio (SNR) is 0.46 dB higher when the COT orients the quantizer.



(a) KLT

(b) COT

Fig. 6. (a) Quantizer oriented with the KLT. (b) Oriented with the COT. Data vectors are indicated with ·'s, and the reproduction vectors are indicated with +'s.

A. Quantizer Optimization

To optimize the quantizers, we adjust the number of coder M, number of reproduction values in each coordinate $\mathcal{K}_J^{(\mathrm{m})}$, and each reproduction value $r_{Ji}^{(\mathrm{m})}$ to minimize the cost function (20). This optimization is most conveniently performed in the local transform coordinates defined by $W^{(\mathrm{m})}$. Rewriting the cost function for transform coder m in terms of transform coefficients $\hat{s}_J^{(\mathrm{m})} = (W^{(\mathrm{m})})^T(x - \mu^{(\mathrm{m})})$ yields

$$\mathcal{C}_{\mathrm{m}} = \frac{1}{N_{\mathrm{m}}} \sum_{J=1}^{d} \sum_{i=1}^{\mathcal{K}_J^{(\mathrm{m})}} \sum_{\hat{s}_J \in R_{Ji}^{(\mathrm{m})}} \left(\left| \hat{s}_J^{(\mathrm{m})} - r_{Ji}^{(\mathrm{m})} \right|^2 - 2\sigma^2 l_{Ji}^{(\mathrm{m})} \right) - 2\sigma^2 \pi_{\mathrm{m}} \log \pi_{\mathrm{m}} \tag{33}$$

where $l_{Ji}^{(\mathrm{m})} = -\log \mathrm{p}(r_{Ji}^{(\mathrm{m})}|\mathrm{m})$ is commonly interpreted as the code word length.

Minimizing the cost (33) with respect to the reproduction values places each one at the mean of the transform coefficients assigned to it

$$r_{Ji}^{(\mathrm{m})} = \frac{1}{N_{Ji}^{(\mathrm{m})}} \sum_{\hat{s}_J \in R_{Ji}^{(\mathrm{m})}} \hat{s}_J^{(\mathrm{m})} \tag{34}$$

where $N_{Ji}^{(\mathrm{m})}$ are the number of transform coefficients in $R_{Ji}^{(\mathrm{m})}$.

For entropy-constrained transform coding, selecting the noise variance σ^2 is equivalent to selecting a target entropy. The target entropy determines the number of transform coders M and the number of reproduction values $\mathcal{K}_J^{(\mathrm{m})}$ in each scalar quantizer. The entropy terms in (33) move the partition away from the minimum mean squared error solution so that reproduction values with low prior probabilities may have no data items assigned to them. Reproduction values with $\mathrm{p}(r_{Ji}^{(\mathrm{m})}|\mathrm{m}) = 0$ can be removed from the coder, reducing the value of $\mathcal{K}_J^{(\mathrm{m})}$. Likewise, coders with low priors may have no data items assigned to them, allowing the number of coders to be reduced. For a recent comprehensive review of quantization methods, see [20].

IV. Algorithm Evaluation

We evaluate our adaptive transform coding algorithm on a database of synthetic aperture radar (SAR) images. We compare compression performance of our method to that of classic transform coders based on the KLT and DCT. We also compare performance to that of KLT- and DCT-based adaptive transform coders. All coders use optimal entropy-constrained quantizers [2]. We report compression performance as signal-to-noise ratio (SNR), in decibels, versus entropy, in bits per pixel (b/pixel). No entropy coding is performed since it introduces variability in performance unrelated to the algorithms.

A. Design Algorithm Summaries

Our adaptive transform coding design algorithm iteratively updates the data partition and the model parameters to optimally compress a training data set.

1) Select the noise variance σ^2. This choice determines the compressed bit-rate.
2) To initialize, select M random data vectors as region means and divide the data space using K-means clustering [8].
3) Iteratively optimize the parameters and partition until the change in cost is negligible.
   a) Update the grid means μ and transforms W according to (27) and (30), respectively.
   b) Transform the data to the appropriate local basis $W^{(\mathrm{m})}$ and update the quantizer reproduction values r according to (34).
   c) Partition the training data to minimize entropy-constrained distortion according to (25) and (26).

In this implementation, we specify the noise variance rather than the compressed bit-rate. This approach makes evaluating performance over a range of bit-rates simple and straightforward. If a particular compressed bit-rate is required, one can select an entropy value and adjust the noise variance to enforce that bit-rate. We took this latter approach in our prior global transform coding work [18].

The adaptive DCT-based transform coders are designed using a generalized Lloyd type algorithm with the transform W constrained to be the DCT. The process is similar to that for the optimal transform coder, but instead of finding the COT at every iteration, we perform a DCT once at the beginning of the optimization process. The process is completed using the resulting transform coefficients.

We also include results for adaptive KLT-based coding, similar to that developed by Effros *et al.* [5] and that presented in our previous work with *fixed-rate* adaptive transform coding [16]. This algorithm is identical to the optimal algorithm above, *except that the transform optimization is replaced with the KLT calculation*. This transform update does not, in general, reduce the cost function. Replacing the COT with the KLT does *not* yield a generalized Lloyd type algorithm unless the data is Gaussian [5]. This design algorithm, as well as the transform coding design algorithms described in [5] and [16], are not guarenteed to converge to a local cost minimum. In practice, we found that the cost almost always increased when the KLT was updated and then decreased when the quantizers and partition were optimized. To handle these frequent cost increases, we monitored the *absolute* change in coding cost and stopped the process when this absolute change became relatively small.

To reconstruct a compressed image, the decompression engine must have the transform coder parameters. The storage space required for the transform coder parameters is referred to as *overhead*. For the tested transform coders, the overhead was 10 bits (three decimal digits) for each transform element and 18 bits (five decimal digits + sign) for each reproduction value and each associated prior probability. Since the DCT is a fixed for all images, it can be hard-coded into the decompression software. The SNR improvement that the COT provides over the DCT is not enough to compensate for the increased overhead. The deleterious effect of overhead on compression performance is greater for adaptive transform coders, making it impractical to develop such coders for individual images. However, adaptive coders can compress databases of related images effectively [4], [5] since the overhead can be amortized over the whole database. As a rule of thumb, an adaptive coder with one local coder per data base image (500 KB each) would have an overhead of less than 1% of the compressed database size (16:1 compression).

B. Evaluation on Image Database

Database compression provides an important and practical application for adaptive transform coding. While the data contained in an individual data file, such as an image, is nonstationary, the characteristics of the different files within the database are often similar. Consequently, one adaptive transform coder can be developed and subsequently used to compress all files within the database. This allows us to incorporate the transform coder parameters into the decompression engine, alleviating the overhead problem. In this section, we compare performance of adaptive and global transform coders developed on a training image and applied to other database images.

We evaluated the adaptive transform coders on a small database (18 MByte) of synthetic aperture radar (SAR) images [21]. Our database consists of 11 images acquired via space-borne radar by the space shuttle [22]. Each image contains three pseudo-color channels: red is L-band (24 cm) horizontally transmitted and received, green is L-band horizontally transmitted and vertically received, and blue is C-band (6 cm) horizontally transmitted and received. Prior to compression, each image is decomposed into its three channels, and the pixels in each channel are divided into 8 × 8 blocks to form

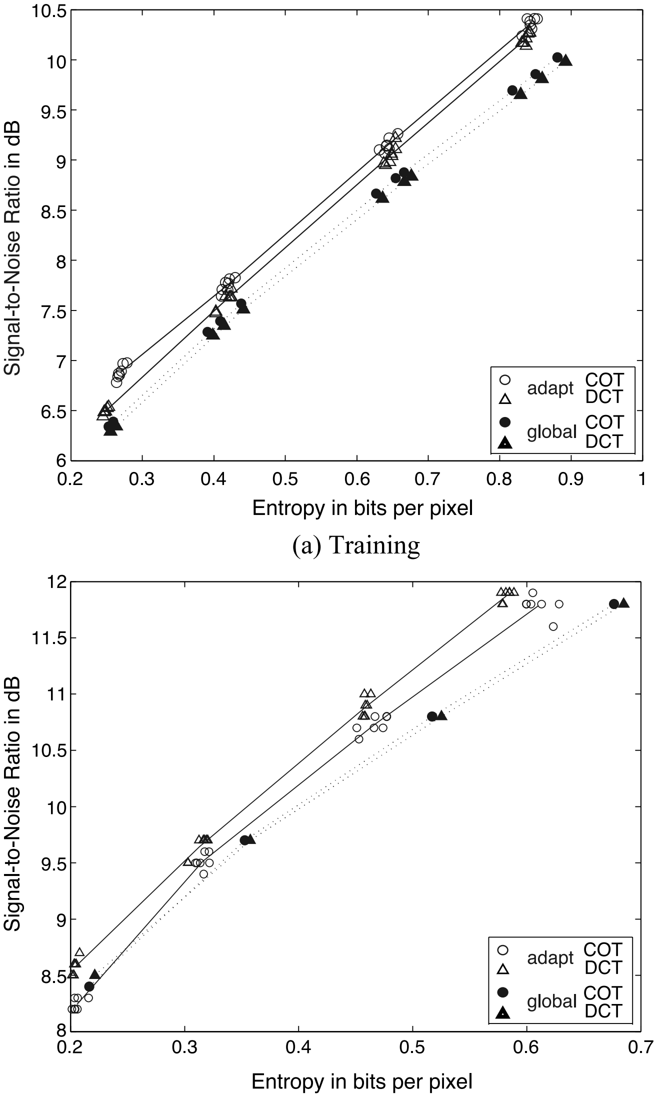
64-dimensional data vectors. SAR images of Belgrade, Taipei, and San Diego constitute the training set (5.9 MBtyes) used to optimize the transform coder parameters. We evaluated compression performance on eight SAR images chosen for their diversity of land uses and terrain types. The test images were acquired over Athens, Boston, Hampton, Honolulu, Laughlin (Colorado River), Lisbon, Phnom Penh, and Ventura.

We developed both global and adaptive transform coders for the image training set at noise variances of 800, 480, 320, and 240. The compressed bit-rates ranged from 0.2 to 0.7 b/pixel. We trained seven adaptive coders at each noise variance starting from different random initializations. The adaptive coders contained 64 regions or local coders. To achieve low bit-rates, the entropy constraint will ensure that some of the local coders are not used, that is, $\pi_{\mathrm{m}} = 0$. In this work, we found that the trained adaptive transform coders used from 37 to 63 of the local coders.

We report SNR results relative to compressed bit-rate given in terms of entropy. All adaptive coder entropies include the rate required to specify the best coder for each image block. For the results presented below, the parameters were included in the decompression engine, so overhead was not included in the bit-rate. The overhead can also be amortized over the whole database, but the effect on the bit-rate depends on the amount of compression. If the 18 Mbyte database were compressed 16:1 (0.5 b/pixel), the overhead due optimal (COT-based) adaptive transform coding increases the compressed bit-rate by 0.028 b/pixel. The increase due to adaptive DCT-based coding is 0.011 b/pixel, due to global COT-based coding 0.004 b/pixel, and to global DCT-based coding 0.0013 b/pixel.

Our results show that a single adaptive transform coder can perform well on a database of related images. Compression results for the training image and the Lisbon image shown in Fig. 7 demonstrate the relative performance of the COT- and DCT-based compression methods. For the training image, adaptive coder SNR is approximately 0.5 dB higher than global coder SNR. The COT-based adaptive coder has an SNR about 0.1 dB higher than the DCT-based adaptive coder. For the Lisbon test image, the global DCT- and COT-based coders have similar performance. The COT-based adaptive coder's SNR is 0.3 dB higher, and the DCT-based adaptive coder's SNR is 0.6 dB higher than the global coder SNR. Note that the adaptive COT-based coder generalizes less well than the adaptive DCT-based coder.

We saw similar results for the eight test images and for the three images included in the training set. At entropies of 0.5 b/pixel, the 11-image SNRs ranged from 8 to 12 dB. The adaptive COT-based coder had higher SNR than the global coders for all but one test images (Ventura) and had an average improvement of 0.2 dB. The adaptive DCT-based coder had higher SNR than the customized global coders for all test images and had an average improvement of 0.4 dB. The KLT-based adaptive coder had test image SNRs comparable with the COT-based adaptive coder. The adaptive DCT-based coder generalized better than the COT-based coder with consistently better SNRs (average 0.2 dB) on the test images.



(a) Training

(b) Lisbon

Fig. 7. Signal-to-noise ratio versus entropy for training and Lisbon images. Compression performance of adaptive coding with 64 local coders to global transform coders. Results are shown for adaptive COT-based coders (open circles), global COT-based coder (solid circles), adaptive DCT-based coders (open triangles), and global DCT-based coder (solid triangles). For the adaptive coders, the lines pass through the means of seven trials.

Another important aspect of compression is time (training time, encoding, or compression time and restoration or decompression time). An adaptive transform coder requires more time for training and encoding than does a comparable global transform coder, although the restoration times are about the same. Training and encoding are done once for the items in a database, making the larger processing time less important than the reconstruction time. Our adaptive COT-based coders required an average training time of 900 min and our adaptive DCT-based coders an average time of 500 min on a 750 MHz Sun Ultra-SPARC III. The global transform coders required 36 min and 4.3 min for the COT and DCT versions, respectively. Adaptive coders also require longer encode times. The adaptive COT-based coder required 352 s to encode the Lisbon image compared with 4.1 s for the global coder. The adaptive DCT-based coder required 76.5 s compared with 3.3 s for the global coder. However, the differences in reconstruction time are small. Adaptive COT-based coders require 3.9 s to decompress the Lisbon image compared with 3.8 s for global COT, 3.2 s for adaptive DCT, and 3.1 s for the global DCT-based coders. The DCT-based transform coders are faster than the COT-based transform coders due to the lower complexity of performing the transform. For d-dimensional data, the complexity for the COT (and KLT) transforms is $\mathcal{O}(d^2)$, whereas the complexity for the DCT is $\mathcal{O}(d \log d)$.

V. Summary

This paper describes our work to develop algorithms for global and adaptive transform coder design. Existing transform coding design algorithms are constructed by concatenating separately designed and often heuristically designed transforms and quantizers. In contrast to this approach, we derived a generalized Lloyd type algorithm for optimal transform coder design starting from a probabilistic framework. A significant and necessary part of this work is a new transform (the COT) that minimizes mean squared error. Definition of this transform made possible our development of an algorithm that integrates optimization of all transform coder parameters: the signal space partition, the transform, and the quantizers. Our new algorithm casts transform coding as a constrained form of vector quantization, allowing systematic development of custom adaptive transform coders and filling a void in the compression literature.

We evaluated our adaptive transform coder on a database of SAR images. Adaptive coders have been referred to as "universal coders" [5] since with enough local coders, they can adapt to a variety of input signals. Our results on the SAR image database indicate that a single adaptive transform coder can be used effectively to compress databases. Adaptive transform coders compressed test images with SNRs approximately 0.5 dB better than global transform coders. COT-based coders have the best SNRs on training images, as expected. However, DCT-based adaptive coders generalize better to new data as they had better test image SNRs than either COT- or KLT-based coders.

Appendix A
COT Derivation

To optimize the transform, we find the orientation of the quantizer grid that minimizes the coding cost. The partition assigns each data vector x to a quantizer reproduction vector $\mathbf{q}_\alpha$ defining subregions R_α. The transform W is constrained to be orthogonal, that is, $W^T W = \mathbf{I}$. The cost function to be minimized is thus

$$\begin{aligned}\mathcal{C} = \frac{1}{N}\sum_{\alpha=1}^{\mathcal{K}}\sum_{x\in R_\alpha}\left\| x - \sum_{J=1}^{d} W_J q_{\alpha J}\right\|^2 \\ + \sum_{K=1}^{d}\sum_{L=1}^{d}\gamma_{KL}\left(W_K^T W_L - \delta_{K,L}\right) \qquad (35)\end{aligned}$$

where W_J is the Jth column vector of W, $q_{\alpha K}$ is the Kth coordinate of reproduction vector $\mathbf{q}_\alpha$, and γ_{KL} is a Lagrange multiplier. The change in cost with respect to an infinitesimal change in one or more elements of W_L is

$$\begin{aligned}\delta\mathcal{C} = \frac{1}{N}\sum_{\alpha=1}^{\mathcal{K}}\sum_{x\in R_\alpha} & \\ \times \Bigg(\left(x - \sum_K W_K q_{\alpha K}\right)^T & (-\delta W_L q_{\alpha L}) \\ + (-\delta W_L q_{\alpha L})^T & \left(x - \sum_K W_K q_{\alpha K}\right)\Bigg) \\ + \sum_K (\gamma_{KL} + \gamma_{LK}) & W_K^T \delta W_L. \qquad (36)\end{aligned}$$

Since W is orthogonal, $W_K^T \delta W_L + \delta W_K^T W_L = 0$. Consequently, the terms containing $q_{\alpha K} q_{\alpha L}$ cancel, and (36) simplifies to

$$\delta\mathcal{C} = \left(\frac{2}{N}\sum_{\alpha=1}^{\mathcal{K}}\sum_{x\in R_\alpha} q_{\alpha L} x^T + \sum_K (\gamma_{KL} + \gamma_{LK}) W_K^T\right)\delta W_L. \qquad (37)$$

At a minimum of the cost, $\delta\mathcal{C}$ is zero. Since the change in W_L is arbitrary, this means the term in parenthesis must be zero. Post-multiplying (37) by W_J and using the orthogonality of W yields

$$\begin{aligned}\sum_\alpha q_{\alpha J}\sum_{x\in R_\alpha} x^T W_K &= -\sigma^2 \frac{\gamma_{JK} + \gamma_{KJ}}{2} \\ &= \sum_\alpha q_{\alpha K}\sum_{x\in R_\alpha} x^T W_J \qquad (38)\end{aligned}$$

or

$$QW = W^T Q^T, \text{ where } Q = \sum_\alpha \mathbf{q}_\alpha \sum_{x\in R_\alpha} x^T. \qquad (39)$$

This symmetry condition (39), along with the orthogonality condition, uniquely defines the COT. The Q matrix is $d \times d$, and the transform W contains $d \times g$ elements, where d is the data dimension, and $g \leq d$ are the number of scalar quantizers with *more* than one reproduction value. Therefore, we require dg equations to uniquely specify W. The symmetry condition $QW = W^T Q^T$ provides $g(g-1)/2 + (d-g)g$ equations, and the orthogonality condition $W^T W = \mathbf{I}$ provides $g(g+1)/2$ equations for the required total of dg equations.

Appendix B
Matrix Rotation Derivation

To minimize distortion, the COT orients the quantizer grid so that the QW matrix is symmetric (30). We can quantify how far the matrix is from symmetric with the sum squared differences between transposed matrix elements

$$A = \sum_{K=1}^{d-1}\sum_{J=K+1}^{d} (a_{KJ} - a_{JK})^2 \qquad (40)$$

where a_{KJ} is the Kth row and Jth column element of QW. We apply Givens rotations [17] $G(K, J, \theta)$ to minimize A. Multiplication by the $G(K, J, \theta)$ matrix applies a rotation of θ rad to the

(K, J) coordinate plane. For a $d \times d$ matrix, there are $(d^2 - d)/2$ such planes.

Applying the rotation $G(K, J, \theta)$ to Q changes the asymmetry A (40) to

$$\begin{aligned}\hat{A} = &\sum_{I \neq K,J} (a_{KI} \cos\theta - a_{JI} \sin\theta - a_{IK})^2 \\ &+ \sum_{I \neq K,J} (a_{KI} \sin\theta - a_{JI} \cos\theta - a_{IJ})^2 \\ &+ (a_{KJ} \cos\theta - a_{JJ} \sin\theta - a_{KK} \sin\theta + a_{JK} \cos\theta)^2. \quad (41)\end{aligned}$$

Minimizing (42) with respect to the rotation angle θ yields

$$\begin{aligned}0 = &\sum_{I \neq K,J} (a_{IK} a_{KI} + a_{IJ} a_{JI}) \sin\theta \\ &+ \sum_{I \neq K,J} (a_{IK} a_{JI} - a_{IJ} a_{KI}) \cos\theta \\ &\times \left((a_{KK} + a_{JJ})^2 - (a_{KJ} - a_{JK})^2\right) \cos\theta \sin\theta \\ &+ (a_{KK} + a_{JJ})(a_{KJ} - a_{JK})(\sin^2\theta - \cos^2\theta). \quad (42)\end{aligned}$$

Substituting $\sin\theta = \tan\theta/\sqrt{(}1 + \tan^2\theta)$ and $\cos\theta = 1/\sqrt{(}1 + \tan^2\theta)$ into (43) yields a solution for θ that is quartic in $\tan\theta$. However, when the rotation angle θ is small, such that $1 + \tan^2\theta \approx 1$, (43) simplifies to

$$\begin{aligned}0 = &\sum_{I \neq K,J} (a_{IK} a_{KI} + a_{IJ} a_{JI}) \tan\theta \\ &+ \sum_{I \neq K,J} (a_{IK} a_{JI} - a_{IJ} a_{KI}) \\ &\times \left((a_{KK} + a_{JJ})^2 - (a_{KJ} - a_{JK})^2\right) \tan\theta \\ &+ (a_{KK} + a_{JJ})(a_{KJ} - a_{JK}). \quad (43)\end{aligned}$$

Solving for $\tan\theta$ yields

$$\tan\theta \approx \frac{(a_{KK} + a_{JJ})(a_{KJ} - a_{JK})}{\Phi} - \sum_{I \neq K,J} \frac{(a_{JI} a_{IK} - a_{JI} a_{IK})}{\Phi} \quad (44)$$

where

$$\Phi = \sum_{I \neq K,J} (a_{JI} a_{IJ} + a_{KI} a_{IK}) + (a_{KK} + a_{JJ})^2 - (a_{KJ} - a_{JK})^2. \quad (45)$$

Acknowledgment

The images used in this work were acquired from the Jet Propulsion Laboratory website. The authors thank the reviewers for their many helpful suggestions and comments.

References

[1] S. Lloyd, "Least square optimization in PCM," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 129–137, Apr. 1982.

[2] N. Farvardin and J. Modestino, "Optimum quantizer performance for a class of non-Gaussian memoryless sources," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 485–497, June 1984.

[3] G. Wallace, "Overview of JPEG (ISO/CCITT) still image compression standard," *Commun. ACM*, vol. 4, no. 4, pp. 30–40, 1991.

[4] R. Dony and S. Haykin, "Optimally adaptive transform coding," *IEEE Trans. Image Processing*, vol. 4, pp. 1358–1370, Oct. 1995.

[5] M. Effros, P. Chou, and R. Gray, "Weighted universal image compression," *IEEE Trans. Image Processing*, vol. 8, pp. 1317–1328, Oct. 1999.

[6] M. Tipping and C. Bishop, "Mixture of probabilistic principal component analyzers," *Neural Comput.*, vol. 11, no. 2, pp. 443–483, 1999.

[7] C. Archer and T. Leen, "Optimal dimension reduction and transform coding with mixture principal components," in *Proc. Int. Joint Conf. Neural Networks*, July 1999.

[8] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Stat. Prob.*, vol. 3, 1967, p. 281.

[9] N. Kambhatla and T. K. Leen, "Optimal dimension reduction by local PCA," *Neural Comput.*, vol. 9, no. 7, pp. 1493–1516, 1997.

[10] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.

[11] P. Chou, T. Lookabaugh, and R. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 31–41, Jan. 1989.

[12] S. Nowlan, "Soft competitive adaptation: Neural network learning algorithms based on fitting statistical mixtures," Ph.D. dissertation, School of Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, 1991.

[13] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc. B*, vol. 39, pp. 1–38, 1977.

[14] S. Roweis and Z. Ghahramani, "A unifying review of linear Gaussian models," *Neural Comput.*, vol. 11, no. 2, pp. 306–345, 1999.

[15] R. Neal and G. Hinton, "A View of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*, M. Jordan, Ed. Boston, MA: Kluwer, 1998.

[16] C. Archer and T. Leen, "From mixtures of mixtures to adaptive transform coding," in *Advances in Neural Information Processing Systems 13*, T. Leen, T. Dietterich, and T. Tresp, Eds. Cambridge, MA: MIT Press, 2001.

[17] G. Golub and C. V. Loan, *Matrix Computations*. Baltimore, MD: John Hopkins Univ. Press, 1989.

[18] C. Archer and T. Leen, "The coding-optimal transform," in *Proc. IEEE Comput. Soc. Data Compression Conf.*, Mar. 2001.

[19] V. Goyal, J. Zhuang, and M. Vetterli, "Transform coding with backward adaptive updates," *IEEE Trans. Inform. Theory*, vol. 46, pp. 1623–1633, July 2000.

[20] R. Gray and D. Neuhoff, "Quantization," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2325–2383, Nov. 1998.

[21] What is Imaging RADAR?, T. Freeman. (1996, Jan.). *http://southport.jpl.nasa.gov/desc/imagingradarv3.html* [Online]

[22] SIR-C/X-SAR Images of Earth, Jet Propulsion Labs.. [Online]. Available: http://www.jpl.nasa.gov/radar/sircxsar

Cynthia Archer received the Ph.D. degree in computer science from the Oregon Health and Science University, Beaverton, in 2002. Her thesis research involved the development of algorithms for data compression and modeling with applications in the areas of fault detection and image processing.

She is currently a research engineer with Research Triangle Institute, Lake Oswego, OR. Her research interests include adaptive signal processing, sensor and image fusion, fault detection, and pattern recognition. Prior to returning to school to earn her Ph.D., she was a design engineer at GTE Government Systems, Needham, MA, where she developed custom digital hardware and embedded real-time software for spread-spectrum satellite communications and radar target tracking equipment.

Todd K. Leen received the Ph.D. degree in theoretical physics from the University of Wisconsin, Madison, in 1982.

He is a Professor of computer science and engineering at the OGI School of Science and Engineering, Oregon Health and Science University, Beaverton. His research interests include machine learning and theoretical neuroscience. His work in machine learning ranges from stochastic search dynamics to local linear techniques and includes applications to coding, data fusion, data and model fusion, fault detection, and signal processing. Prior to his current academic career, he was a scientist/engineer at IBM Burlington, VT. He is an editor for the journal *Neural Computation*.

Dr. Leen sits on the NIPS Foundation board of directors