An Algorithm for Learning Orthonormal Matrix Codebooks for Adaptive Transform Coding

Rashmi Boragolla ¹ and Pradeepa Yahampath ²

¹University of Manitoba
²Affiliation not available

October 30, 2023

Abstract

This paper proposes a novel data-driven approach to designing orthonormal transform matrix codebooks for adaptive transform coding of any non-stationary vector processes which can be considered locally stationary. Our algorithm, which belongs to the class of block-coordinate descent algorithms, relies on simple probability models such as Gaussian or Laplacian for transform coefficients to directly minimize with respect to the orthonormal transform matrix the mean square error (MSE) of scalar quantization and entropy coding of transform coefficients. A difficulty commonly encountered in such minimization problems is imposing the orthonormality constraint on the matrix solution. We get around this difficulty by mapping the constrained problem in Euclidean space to an unconstrained problem on the Stiefel manifold and leveraging known algorithms for unconstrained optimization on manifolds. While the basic design algorithm directly applies to non-separable transforms, an extension to separable transforms is also proposed. We present experimental results showing that adaptive coding with our transform codebook designs outperform the commonly used discrete cosine transform (DCT) in many regions of natural images and motion-compensated prediction error images of video frames. According to these results, the proposed codebook designs slightly outperform those found by a common alternative approach, the sparsity-based optimization.
Abstract—This paper proposes a novel data-driven approach to designing orthonormal transform matrix codebooks for adaptive transform coding of any non-stationary vector processes which can be considered locally stationary. Our algorithm, which belongs to the class of block-coordinate descent algorithms, relies on simple probability models such as Gaussian or Laplacian for transform coefficients to directly minimize with respect to the orthonormal transform matrix the mean square error (MSE) of scalar quantization and entropy coding of transform coefficients. A difficulty commonly encountered in such minimization problems is imposing the orthonormality constraint on the matrix solution. We get around this difficulty by mapping the constrained problem in Euclidean space to an unconstrained problem on the Stiefel manifold and leveraging known algorithms for unconstrained optimization on manifolds. While the basic design algorithm directly applies to non-separable transforms, an extension to separable transforms is also proposed. We present experimental results showing that adaptive coding with our transform codebook designs outperform the commonly used discrete cosine transform (DCT) in many regions of natural images and motion-compensated prediction error images of video frames. According to these results, the proposed codebook designs slightly outperform those found by a common alternative approach, the sparsity-based optimization.

Index Terms—Adaptive transform coding, quantization, Stiefel manifold, constrained optimization.

I. INTRODUCTION

Transform coding [1] is a simple, yet highly effective method for compression of correlated signals, wherein a linear orthogonal transformation is applied to vectors of signal samples prior to quantization. By using only scalar quantization on transform coefficients, transform coding can achieve rate-distortion performance equivalent to vector quantization of the original vector. This apparent coding gain is due to the fact that a proper transformation of a correlated vector can produce an uncorrelated vector with most of the energy “compacted” into a few transform coefficients and the accurate scalar quantization of these few coefficients is sufficient to obtain a good approximation to the original vector. Transform coding is the basis for all modern image and video compression algorithms [2], [3].

The performance of a transform code heavily depends on the choice of the orthogonal transform, and the optimal transform can in general depend on the distribution of the source vector and the quantization bit-rate. It is well-known that, regardless of the bit rate, the optimal transform for a stationary Gaussian source under the MSE criterion is the Karhunen-Loève transform (KLT) [1], [4] which is a function of the covariance matrix of the source vector. In particular, the KLT is the transform that results in the highest energy compaction for Gaussian vectors. However, this elegant result cannot be straightforwardly applied to real-world signals as they are neither Gaussian nor stationary. For non-Gaussian signals, KLT maybe not be optimal [5]. Even if the Gaussian assumption holds, the covariance matrix of a non-stationary source varies across the signal. The optimal transform coding of a non-stationary signal requires the adaptation of the transform matrix to the local statistics of the signal. In general, transform adaptation can be realized into two fundamentally different ways, forward adaptation or backward adaptation [1, Ch. 16]. In forward adaptation, the transform matrix is adapted to the input signal periodically at the encoder, for example using a codebook of transform matrices, and the codebook index is explicitly signaled to the decoder.

This paper makes a new contribution toward designing codebooks of orthonormal transform matrices for forward adaptation. We present a new data-driven learning algorithm for optimizing a fixed-size codebook of orthogonal transform matrices for coding any non-stationary vector process which can be well modeled by a block-wise stationary process. This algorithm belongs to the class of block coordinate-descent (BCD) algorithms [7] and in spirit, is similar to the generalized Lloyd algorithm for vector quantizer design [1]. Our algorithm can be applied to any non-stationary signal, so long as a simple probability model for transform coefficients is available. If the signal samples are approximately jointly Gaussian, then the transform coefficients will be close to Gaussian as well. This scenario is likely to occur in most real-world signals. The Laplace distribution for transform coefficients is also a good practical approximation in such scenarios. In particular, the Laplace model is commonly used in the contexts of image and video signals [8]–[10]. We present analytical developments required to apply the algorithm for the Gaussian model under high-rate assumptions [1] and the Laplace model for any rate. The algorithm can be applied to encoders with both uniform and non-uniform quantizers. We have focused on uniform scalar quantizers with a dead-zones and entropy coding, which is the most prevalent approach in practice [11].

The algorithm is directly applicable to the design of non-separable transforms, but we also present an extension to the design of separable transforms. A novel aspect of our approach to matrix codebook design is the way we handle the orthonormality constraint. Rather than minimizing an objective $J(T)$ subject to an orthonormality constraint on the matrix argument $T$ in Euclidean space, we map the problem to a simpler unconstrained problem on Stiefel
manifold \([12]\). This approach allows us to explicitly minimize
the transform coding MSE with respect to an orthonormal
transform matrix, which is in contrast to other indirect methods
used to enforce orthonormality constraints in similar problems,
such as Givens parameterization \([13]\) \([15]\), (see \([16]\) for a
general discussion). With matrix-parameterization, the adap-
tation codebook consists of quantized parameters rather than
the orthonormal matrices themselves, which is inefficient in a
rate-distortion sense.

We have applied the proposed algorithm to design adaptive
transform codebooks for still images and motion compensated
video prediction errors, and compared the results with those produced by non-adaptive coding based on the commonly used
two-dimensional DCT (2D-DCT) \([2]\) as well as adaptive cod-
ing using transform codebooks designed with recently reported
sparse orthonormal transforms (SOT) \([17]\). The SOT method
is a pursuit algorithm for minimizing the approximation MSE
subject to an \(l_0\)-sense sparsity constraint on the transform
coefficient vector. This method does not take into account
any quantizer characteristics, but rather forces the transforms
to produce coefficient vectors with a sparsity level that is
determined by an \(l_0\) penalty on the linear approximation MSE.
Our algorithm is fundamentally different in that it directly
minimizes the quantization MSE of the transform coefficients
for a given quantizer structure (such as uniform quantization
with a given step-size). Furthermore, with the SOT algorithm,
the Lagrangian weight required to achieve a given sparsity
level has to be found through an annealing procedure used in
conjunction with the iterative pursuit algorithm. Our algorithm
only requires the specification of the step-size of uniform
quantizers to be applied to the transform coefficients. In
our experiments with image and video data, the transform
codebooks produced by the proposed method were observed
to be slightly better than those obtained with the SOT method.

Related work: In addition to \([17]\), many other approaches to
transform codebook construction have been recently reported
in the literature \([13]\), \([18]\)–\([24]\). However, unlike our method
which is applicable to any signal (images or otherwise), these
methods are highly application dependent and rely on specific
image structures such as the presence of oriented edges. Some
of the very early work on adaptive transforms include \([25]\)–
\([30]\). In \([25]\), \([26]\), non-stationary image data is classified into
classes and the KLT estimated from class data is used as the
transform for that class, where the classification and KLT
estimation are iterated. \([27]\) and \([28]\) follow a similar idea but
use trained fully-connected neural networks for classification
into classes. In \([30]\) a Gaussian mixture model for source data is used to iteratively optimize a transform codebook and quantizer reproduction levels. However, this Gaussian-
mixture-based transform code representation is inefficient for
adaptive coding as it requires encoding other side information
in addition to the transform matrices.

The rest of the paper is organized as follows. Section \([\text{II}]\)
defines the transform coding problem for block-wise stationary
signals. Section \([\text{III}]\) presents the main transform codebook op-
timization algorithm proposed in this paper for non-separable
transforms. The extension of this algorithm to separable trans-
forms is given in Section \([\text{IV}]\). Experimental results obtained
with image and video data are presented in Section \([\text{V}]\). The
paper is closed with conclusions in Section \([\text{VI}]\).

Notation: Bold symbols denote matrices and vectors. \((\cdot)^T\)
and \((\cdot)^H\) respectively denote the transpose and conjugate-
transpose of a matrix or vector. \(I_K\) denotes the \(K \times K\)
identity matrix. \(e_j\) denotes the \(j\)th column of an identity matrix
whose size is to be implied by the context. \(E[\cdot]\) denotes the
expectation operator. \(A \otimes B\) denotes Kronecker product of
matrices \(A\) and \(B\). \(\text{vec}(A)\) is the vector obtained by stacking
the columns of the matrix \(A\). If \(X = \text{vec}(A)\), then we write
\(A = \text{vec}^{-1}(X)\).

II. TRANSFORM CODING OF RANDOM VECTORS

A. Stationary vectors

Consider transform coding of a zero-mean stationary random
vector \(\mathbf{X} \in \mathbb{R}^K\). The encoding process consists of two steps. In the first step, \(\mathbf{X}\) is linearly transformed into an equal
size vector

\[
\mathbf{Y} = \mathbf{T} \mathbf{X},
\]

where \(\mathbf{T} \in \mathbb{R}^{K \times K}\) is an orthonormal matrix. In the second
step, the transform coefficients \(\mathbf{Y} = (Y_1, \ldots, Y_K)^T\) are scalar
quantized. In this paper, we consider uniform quantization and
entropy coding of transform coefficients, which is common
practice in most applications, including image and video com-
pression. Further, for simplicity of presentation, we assume
that the same quantization step-size \(\Delta\) is used for all coeffi-
cients. However, our formulation can be easily generalized to
the case of unequal step sizes. Let the quantized versions of
\(\mathbf{X}\) and \(\mathbf{Y}\) be \(\hat{\mathbf{X}} = (\hat{X}_1, \ldots, \hat{X}_K)^T\) and \(\hat{\mathbf{Y}} = (\hat{Y}_1, \ldots, \hat{Y}_K)^T\)
respectively. The transform coded source vector is given by
\(\hat{\mathbf{X}} = \mathbf{T}^{-1} \hat{\mathbf{Y}}\). We assume that the MSE of a uniform quantizer
is a function of its step-size and the input variance \([1]\).
The quantizer input variances \(\sigma^2_k = E[Y_k^2], k = 1, \ldots, K\)
are the diagonal elements of \(\mathbf{C}_Y = \mathbf{T} \mathbf{C}_X \mathbf{T}^T\), where \(\mathbf{C}_X = E[\mathbf{X} \mathbf{X}^T]\)
is the covariance matrix of \(\mathbf{X}\). Accordingly, let the MSE of
quantizing the coefficient \(Y_k\) using the step-size \(\Delta\) be given by
a function \(\theta_k(T, \mathbf{C}_X, \Delta) = E(Y_k - \hat{Y}_k)^2\). The MSE of
transform coding \(\mathbf{X}\) is given by

\[
\Theta(T, \mathbf{C}_X, \Delta) = E\|\mathbf{X} - \hat{\mathbf{X}}\|^2 = \sum_{k=1}^{K} \theta_k(T, \mathbf{C}_X, \Delta).
\]

The bit rate of transform coding \(\mathbf{X}\) is determined by the choice of
quantization step-size \(\Delta\). Suppose that optimal lossless
coding is applied to quantizer outputs so that the minimum
rate of transform coding is the entropy of the quantization
indices. Let \(r_k(T, \mathbf{C}_X, \Delta)\) be the entropy (in bits) of \(\hat{Y}_k\),
such that the total average bit rate required to code \(\mathbf{Y}\) is
\(R(T, \mathbf{C}_X, \Delta) = \sum_k r_k(T, \mathbf{C}_X, \Delta)\). Let there be an average
bit rate constraint of \(R_0\) bits/vector. For a given \(\mathbf{C}_X\) and \(\Delta\),
the optimal orthonormal \(\mathbf{T}\) which satisfies the rate constraint
can thus be found by solving

\[
T^* = \arg \min_{\mathbf{T} \in \mathbb{R}^{K \times K}} J(T, \mathbf{C}_X, \Delta)
\]

subject to \(\mathbf{T}^T \mathbf{T} = I_K\).
\begin{equation}
J(T,C_X,\Delta) = \Theta(T,C_X,\Delta) + \lambda R(T,C_X,\Delta)
\end{equation}

where 

\begin{align*}
\lambda &\geq 0 \text{ is to be chosen to satisfy } R(T,C_X,\Delta) \leq R_0. \\
\text{The solution of } (2) \text{ requires the specification of suitable expressions for } \theta_k(\cdot) \text{ and } r_k(\cdot) \text{ which would depend on the quantizer structure. This issue will be considered in detail in Section III.}
\end{align*}

It is easy to demonstrate that if \( X \) is Gaussian, the solution to (2) is the KLT regardless of the bit rate [1], [4]. However, in general, the solution may be rate-dependent. Note that, a given distortion-rate point can be achieved in two ways: either by letting \( \lambda = 0 \) and searching for \( \Delta \) to meet the rate constraint, or by fixing \( \Delta \) and searching for \( \lambda \). It is typical in practical applications to use rate control algorithms to set the bit rate by varying \( \Delta \) [51].

### B. Non-stationary vectors (adaptive coding)

Next, consider coding a sequence of non-stationary random vectors. In this case, the optimal transform may vary with the input vector, and adaptive coding can significantly improve the rate-distortion performance. While most real-world signals are non-stationary, they can often be well modeled by block-wise or locally stationary signals where the statistical properties such as the mean value and the covariance function remain constant over blocks of consecutive samples. Let \( B \subset S_B \) be a locally stationary block of vectors in the non-stationary signal to be coded, where \( S_B \) is the set of all possible (ensemble of) blocks. We can determine the optimal transform matrix \( T^* \) for each block \( B \) in the ensemble by solving (2) on a per-block basis. For example, if \( X \subset B \) is a Gaussian vector, then the optimal transform for a stationary block \( B \) is the KLT of \( X \subset B \) [1], [4]. However, using \( T^* \) on a per block basis would entail a substantial bit rate overhead to signal the transform matrices to the decoder. Instead, our objective is to use a pre-designed, size-\( N \) codebook of orthonormal transform matrices \( T = \{T_1, T_2, ..., T_N\} \) from which the best transform matrix for an entire block \( B \) is chosen. Specifically, the optimal transform matrix for some given block of vectors, \( B \) is chosen as

\begin{equation}
T^*_B = \arg \min_{T \in T} \frac{1}{|B|} \sum_{X \in B} \|X - \hat{X}(T)\|^2,
\end{equation}

where \( \hat{X}(T) \) is the transform coded version of source vector \( X \) using the transform \( T \), and \(|B|\) is the number of vectors in the block \( B \). The bit rate overhead in this case can be controlled by the choice of the codebook size \( N \), as well as the size of a stationary block.

In the block-stationary model, the covariance matrix of the block of vectors \( B, C_B = E[XX^T|X \in B] \) is random matrix. In this case, we design the transform codebook to minimize the average cost

\begin{equation}
E[J(T,C_B,\Delta)] = E[\Theta(T,C_B,\Delta)] + \lambda E[R(T,C_B,\Delta)],
\end{equation}

where the expectation is taken over the distribution of \( C_B \).

### III. Transform Matrix Codebook Optimization

The process of designing an optimal codebook of size \( N \) for a given source ensemble \( S_B \) essentially involves two steps, namely 1) partitioning \( S_B \) into \( N \) non-overlapping subsets \( (S_1, ..., S_N) \), and 2) assigning a unique transform matrix \( T_i \) to \( S_i \), \( i = 1, ..., N \), such that the cost of transform coding averaged over \( S_B \), \( E[J(T(B),C_B,\Delta)] \) is minimized, where \( T(B) = T_i \) if \( B \subset S_i \). Let \( S_C = \{C_B : B \subset S_B\} \) be the ensemble of block covariance matrices. An equivalent codebook design problem is to partition \( S_C \) into \( N \) non-overlapping subsets \( \mathcal{G} = \{\Omega_1, ..., \Omega_N\} \), where \( \Omega_i \subset S_C \), and to determine the optimal codeword \( T_i \) for each \( \Omega_i \). Assume that \( C_B \) is distributed over \( S_C \) according to some probability distribution. Then, the cost of transform coding all stationary blocks \( B \subset S_B \) using the transform codebook \( T \) is

\begin{equation}
\tilde{J}(T,\mathcal{G}) = \sum_{i=1}^{N} E[J(T_i, C_B, \Delta) | C_B \in \Omega_i] P(\Omega_i),
\end{equation}

where \( P(\Omega_i) = P_B(C_B \in \Omega_i) \) and the expectation is taken over the distribution of \( C_B \). We wish to find the codebook \( T^* \) which minimizes (4) with respect to the sets \( (T, \mathcal{G}) \). Specifically, we seek to solve

\begin{equation}
\text{Minimize } \tilde{J}(T,\mathcal{G})
\end{equation}

subject to \( T^*_i T_i = I_K \), \( i = 1, ..., N \).

The direct solution of this constrained minimization problem appears intractable. The problem is reminiscent of codebook design in vector quantization, where a BCD algorithm, commonly referred to as the generalized Lloyd’s algorithm, is used to solve a similar problem [1]. However, in that case there are no constraints on code vectors. In this paper, we present a BCD algorithm incorporating orthogonal constraints on codebook elements to solve (5), where one alternates between the solutions to two sub-problems described below, such that the algorithm converges to a local minimum of (5). The complete design algorithm is presented in Sec. III-C.

#### A. Sub-problem 1: Optimal partition \( \mathcal{G} \) for a fixed codebook \( T \)

It is straightforward to argue that, for a fixed \( T \), the optimal \( \mathcal{G} \) that minimizes \( J(\mathcal{T}, \mathcal{G}) \) is given by \( \mathcal{G}^* = \{\Omega_1^*, ..., \Omega_N^*\} \), where

\begin{equation}
\Omega_i^* = \{C_B \subset S_C : J(T_i, C_B, \Delta) < J(T_j, C_B, \Delta) \forall j \neq i\}
\end{equation}

with ties broken suitably. Note that the matrix orthogonality constraint is irrelevant to this sub-problem.

#### B. Sub-problem 2: Optimal codebook \( T \) for a fixed partition \( \mathcal{G} \)

Given a partition \( \mathcal{G}, \tilde{T}_i \) only affects the \( i \)-th term of the sum in (4). Let \( \tilde{J}_i(T) = E[J(T, C_B, \Delta) | C_B \in \Omega_i] \) (for notational simplicity we will ignore the dependence of \( \tilde{J}_i \) on \( \Delta \). Thus,
Algorithm 1 (Steepest descent on O(K) using Armijo step-size)

Choose $T \in \mathbb{R}^{K \times K}$ such that $T^T T = I_K$. Set step size $\gamma \leftarrow 1$.

1. Compute $D_T [\bar{J}_i(T)]$ and the descent direction $Z = T \left( D_T [\bar{J}_i(T)] \right)^{-1} T - D_T [\bar{J}_i(T)]$.
2. Evaluate $z = \frac{1}{\gamma} \text{tr} \{ Z^T Z \}$. If $\sqrt{z}$ is sufficiently small, stop.
3. If $\bar{J}_i(T) - \bar{J}_i(\pi(T + 2\gamma Z)) \geq 1/2 \gamma z$, then set $\gamma \leftarrow 2\gamma$, and repeat Step 3.
4. $\bar{J}_i(T) - \bar{J}_i(\pi(T + \gamma Z)) < 1/2 \gamma z$ then set $\gamma \leftarrow 2^{3-\iota} \gamma$, and repeat Step 4.
5. Set $T \leftarrow \pi(T + \gamma Z)$
6. Go to Step 1.

The codebook $T^\ast = \{ \hat{T}^\ast_1, \hat{T}^\ast_2, ..., \hat{T}^\ast_N \}$ that minimizes $J(T, G)$ is given by

$$
\hat{T}^\ast_i = \arg \min_{T \in \mathbb{R}^{K \times K}} J_i(T)
$$

subject to $T^T T = I_K$, $i = 1, \ldots, N$.

While the solution to this constrained minimization problem does not appear to be straightforward, we note that the solution space, the set of all $K \times K$ real orthogonal matrices, is the compact manifold referred to as the orthogonal group $O(K) = \{ T \in \mathbb{R}^{K \times K} : T^T T = I_K \}$ [12]. That is to say that, every orthonormal matrix corresponds to a unique “point” on $O(K)$. Therefore, rather than solving (1) as a constrained minimization problem in the Euclidean space $\mathbb{R}^{K \times K}$, we can equivalently solve it as an unconstrained minimization problem on $O(K)$, i.e.,

$$
\hat{T}^\ast_i = \arg \min_{T \in O(K)} \bar{J}_i(T).
$$

Optimization on manifolds is a widely studied problem, see [12] and references therein. Of particular interest to us is the modified steepest-descent algorithm [32 Sec. V-A] which can be used to minimize any differentiable function on the complex Stiefel manifold $\text{St}(K, P) = \{ T \in \mathbb{C}^{K \times P} : T^H T = I_P \}$, the set of all $K \times P$ complex matrices whose columns are orthonormal vectors. This algorithm, which incorporates the Armijo’s step-size rule, almost always converges to a local minimum. As $O(K)$ is a special case of $\text{St}(K, P)$, we have adapted [32 Algorithm 15] in this paper to solve (6). The only requirement is that $\bar{J}_i(T)$ be at least once differentiable with respect to $T$. The adaptation of [32 Algorithm 15] to solve our problem is summarized in Algorithm 1 for any $\bar{J}_i(T)$, where $D_T [F(T)] \in \mathbb{R}^{K \times K}$ denotes the matrix of derivatives of a function $F(T) : \mathbb{R}^{K \times K} \to \mathbb{R}$ with respect to the elements of $T \in \mathbb{R}^{K \times K}$, see [32 Eqn. (2)]. The projection operator [32 Proposition 7] is given by $\pi(W) = UV^T$, where $U$ and $V$ are the matrices whose columns are the left and right singular vectors respectively of any non-singular $W \in \mathbb{R}^{K \times K}$.

Consider

$$
D_T [\bar{J}_i(T)] = E \left[ D_T [\Theta(T, C_B, \Delta)] \right| C_B \in \Omega_i] + \lambda E \left[ R(T, C_B, \Delta) \right| C_B \in \Omega_i] = E \left[ \sum_{k=1}^K D_T [\theta_k(T, C_B, \Delta)] \right| C_B \in \Omega_i] + \lambda E \left[ \sum_{k=1}^K D_T [r_k(T, C_B, \Delta)] \right| C_B \in \Omega_i].
$$

Thus, the requirement that $\bar{J}_i(T)$ be differentiable with respect to $T$ can be met by choosing an appropriate statistical model for the transform coefficients so that it is possible to obtain closed-form differentiable expressions for $\theta_k(T, C_B, \Delta)$ and $r_k(T, C_B, \Delta)$. In coding of image, video, and other real-world signals, transform coefficients are typically modeled by unimodal pdfs such as Gaussian or Laplace pdfs [9], [33]–[35]. In the following, we obtain closed-form expressions for $D_T [\theta_k(T, C_B, \Delta)]$ and $D_T [r_k(T, C_B, \Delta)]$ for the cases of Gaussian or Laplace distributed transform coefficients.

Let $Y(T) = TX$ be the coefficient vector obtained by transforming $X \in B$ using the transform matrix $T$, and let the variances of these transform coefficients be $\sigma^2_{Y_k}(T)$, $k = 1, \ldots, K$, which are the diagonal elements of the matrix $T_C B T^T$. We note that

$$
\sigma^2_{Y_k}(T) = e_k^T T C_B T^T e_k.
$$

1) High-rate Gaussian model: For Gaussian variables, quantization MSE and index entropy have closed-form expressions only when the rate is large. It can be shown that, under high-rate conditions [1], [4], the MSE $\theta_k(T, C_B, \Delta) = \frac{\Delta^2}{12}$ and the entropy [36]

$$
r_k(T, C_B, \Delta) = \frac{1}{2} \log_2 \left( 2\pi e \sigma^2_{Y_k}(T) \right) - \log_2 \Delta.
$$

For fixed $\Delta$, $D_T [\theta_k(T, C_B, \Delta)] = 0$ and $\lambda$ can be any positive constant. It follows that $\frac{d}{dT} \sigma^2_{Y_k}(T) = 2e_k e^T T C_B$ [37], and therefore

$$
D_T [r_k(T, C_B, \Delta)] = \frac{1}{\ln(2)} \frac{e_k e^T T C_B}{\sigma^2_{Y_k}(T)}.
$$

2) Finite-rate Laplacian model: If the transform coefficients are Laplace distributed, it is possible to obtain differentiable closed-form expressions for both the MSE and the entropy of quantizer outputs without high-rate assumptions and therefore applicable for any $\Delta$. We will consider a general form uniform quantizer which has a dead zone $(-\frac{\Delta}{2}, \frac{\Delta}{2})$ and step-size $\Delta$. This type of quantizers are widely used in image and video coding [8]. It is straightforward to show that the MSE of quantizing a mean-zero Laplace variable with variance $\sigma^2$ is

$$
g(b) = 2b^2 - e^{-\frac{b^2}{\sigma^2}} \left( \frac{b^2 - \Delta^2}{4} + zb + \Delta b \left( \frac{e^{\frac{\Delta}{2}} + 1}{e^{\frac{\Delta}{2}} - 1} \right) \right).
$$
where \( b = \sqrt{\frac{2}{\pi}} \) and the entropy of the quantizer output is

\[
h(b) = - (1 - e^{-\hat{\pi}}) \log_2 (1 - e^{-\hat{\pi}}) - e^{-\hat{\pi}} \log_2 \left( \frac{e^{-\hat{\pi}}(1 - e^{-1})}{2} \right) + \frac{\Delta e^{-\hat{\pi}} e^{\hat{\pi}}}{b(e^{\hat{\pi}} - 1) \ln(2)},
\]

where we have assumed that the quantizer has infinite support. For quantizers operating on transform coefficients we thus have \( \theta_k(T) = g(b_k(T)) \) and \( r_k(T) = h(b_k(T)) \) where \( b_k(T) = \sqrt{\frac{1}{\sigma_k^2(T)}} \) with \( \sigma_k(T) \) being given by (9), \( k = 1, \ldots, K \). The expressions for \( D_T [\theta_k(T)] \) and \( D_T [r_k(T)] \) for this case are given by (14) and (15) in Appendix A.

C. Algorithm for transform codebook design

The complete design procedure is presented in Algorithm 2. One starts with a suitable (application dependent) segmentation of a training set of source samples into non-overlapping, locally stationary blocks \( B_1, B_2, \ldots \) such that each block can be treated as a set of vectors with a common covariance matrix. The algorithm is then initialized by estimating the covariance matrices \( C_{B_1}, C_{B_2}, \ldots \) of the stationary blocks. Algorithm 2 then iteratively improves a suitably chosen initial codebook of orthonormal matrices, until the convergence criteria are met. Since the solutions to sub-problem 1 and sub-problem 2 (steps 1 and 2 in the algorithm) can never increase the objective function, the algorithm is guaranteed to converge to at least a local minimum.

D. A toy example

While the KLT is the optimal transform for coding Gaussian vectors, it is known that the KLT is not optimal for vectors from a Gaussian mixture \( \mathcal{N}(\mu, \Sigma) \). The proposed algorithm was used to optimize transform codebooks for 2-dimensional vectors drawn from a Gaussian mixture with three equiprobable mean-zero components \( \mathcal{N}_1, \mathcal{N}_2, \) and \( \mathcal{N}_3 \) whose covariance matrices respectively are

\[
\begin{bmatrix}
1.54 & -1.84 \\
-1.84 & 2.62 \\
\end{bmatrix}, \begin{bmatrix}
0.46 & 0.40 \\
0.40 & 0.70 \\
\end{bmatrix}, \text{ and } \begin{bmatrix}
2.22 & 0.77 \\
0.77 & 0.38 \\
\end{bmatrix}.
\]

Fig. 1(a) shows the directions of a single optimized transform for the Gaussian mixture (a size 1 codebook) super-imposed on data from the Gaussian mixture. Shown here are the transforms obtained using the high-rank Gaussian model \( T_{Gauss} \) and the finite-rank Laplacian model \( T_{Laplace} \). Also shown are the directions of the KLT computed using a single covariance matrix estimated from the complete training set of 3,000,000 source vectors from the Gaussian mixture. Further, Fig. 1(b) shows the directions of the optimized transforms in a codebook of size \( N = 2 \) obtained using the finite-rank Laplacian model. Fig. 2 compares the MSE of codebooks of sizes \( N = 1, 2, 3 \) designed using the Laplacian model, with those of the KLT and the DCT. The MSE curves in Fig. 2 were obtained by testing codebooks optimized for \( \lambda = 0 \) and \( \Delta = 3 \), using quantizers with step-sizes \( \Delta \) in the range 1-10. It was observed that the MSEs of codebooks specifically optimized for each \( \Delta \) in the range shown here nearly coincided with the MSE of a single codebook optimized for \( \Delta = 3 \). Furthermore, in this case, both the high-rank Gaussian model and the finite-rank Laplacian model produced practically the same transforms. Though it is not known if the optimized transforms found by the proposed algorithm are optimal for the Gaussian mixture source, our results show that they are better than the KLT.

IV. Extension to separable transforms

The computational complexity (number of required multiplications and additions) of the transform given by (11) is \( O(K^2) \) for a size \( K \) source vector. In applications such as image and video compression, where the source vectors are formed by taking pixels from rectangular or square blocks, \( K^2 \) can be large, and the resulting computational burden can be quite high. In such applications, it is common to use separable transforms \( [\delta] \) to reduce the dimensionality of the problem. Given a \( L \times L \) matrix \( X' \in \mathbb{R}^{L \times L} \) of source samples, a separable transform is defined by a pair of orthonormal transform matrices \( T_r \in O(L) \) (row transform) and \( T_c \in O(L) \) (column transform), such that the transform coefficient matrix \( Y' \in \mathbb{R}^{L \times L} \) is given by

\[
Y' = T_rX'T_c^T,
\]

By letting \( X = \text{vec}(X') \) and \( Y = \text{vec}(Y') \), (12) can be written in the form (11), where \( K = L^2 \) and \( T = T_r \otimes T_c \). While this formulation allows us to cast the separable problem in the same framework as in Sec. III due to the separability constraint we can no longer perform the steepest-descent on \( O(K) \) with respect to \( T \) to solve the sub-problem 2 (Sec. III-B). The admissible set of \( T \) in this case is only a subset of \( O(K) \), and the optimal solution to sub-problem 2 is now given by

\[
\hat{T} = \arg \min_{T \in O(K)} J(T), \quad T \in O(K),
\]

where \( \hat{O}(K, L) = \{ T \in O(K) : T = T_r \otimes T_c, T_r \in O(L), T_c \in O(L) \} \). Note however that for fixed \( T_c \) (or \( T_r \) \( (13) \) is a minimization problem on \( O(L) \) with respect to \( T_r \) (or \( T_c \)). Based on this observation, we propose two possible adaptations of the Algorithm 1 to solve (13).
Laplacian model. Also shown is the KLT of the Gaussian mixture for comparison.

Fig. 1. Geometric representation of the optimized transform codebooks for a 3 component 2-dimensional Gaussian mixture source \((X_1, X_2)\). (a) Size \(N = 1\) codebook found using the Gaussian model \((T_{Gauss})\) and the Laplacian model \((T_{Laplace})\), and (b) Size \(N = 2\) codebook \(\{T_1, T_2\}\) found using the Laplacian model. Also shown is the KLT of the Gaussian mixture for comparison.

In this section we present experimental results obtained with two types of common real-world signals, gray-scale still images and motion-compensated prediction error (MCPE) images produced in a video encoder. We compare the performance of our codebook designs, which we will refer to as quantization-optimized orthogonal transform (QOT), with the non-adaptive 2D-DCT and the adaptive codebook designs obtained with an alternative approach, the SOT [17]. The total bit rate of an adaptive transform coder is the sum of the bit rate of lossless coding of quantization indices of transform coefficients and the bit rate required to encode the indices representing the transform matrix of each locally stationary block (SB) of pixels. In the following, we will refer to the latter, which is the rate penalty due to transform adaptation, as the adaptation bit rate overhead (ABRO). Throughout, we will use the estimated entropy of the index sequences as a surrogate for the bit rate resulting from lossless coding. The reconstruction quality of images and video frames is most commonly measured in terms of the peak signal-to-noise ratio.

2) Alternating minimization: Iteratively solve the following two problems, each using Algorithm 1 until \(T_i = T_{r_i} \otimes T_{c_i}\) converges to a fixed matrix.

(i) For fixed \(T_{c_i}\), solve 
\[
\hat{T}_{r_i}^* = \arg \min_{T_{r_i}} \bar{J}_i(T_{r_i} \otimes T_{c_i})
\]
\[
T_{r_i} \in O(L)
\]

(ii) For fixed \(T_{r_i}\), solve 
\[
\hat{T}_{c_i}^* = \arg \min_{T_{c_i}} \bar{J}_i(T_{r_i} \otimes T_{c_i})
\]
\[
T_{c_i} \in O(L)
\]

In our experiments, we found that both methods yielded codebooks that have nearly identical transform coding performance, though method 2) converged faster in most cases. Appendix B [see (19), (21), (24), (25), (28), and (29)] presents the derivative expressions required for steepest-descent.

V. EXPERIMENTAL RESULTS

Appendix B [see (19), (21), (24), (25), (28), and (29)] presents the derivative expressions required for steepest-descent.

1) Alternating descent steps: Initialize the Algorithm 1 with some orthonormal \((T_r, T_c)\) pair. In each iteration, (i) fix \(T_c\) and perform a steepest-descent update (steps 1-5) for \(T_r\) in the direction of 
\[
Z_r = T_r (D_T \cdot \bar{J}_i(T_r)) \cdot T_r - D_T \cdot \bar{J}_i(T_r)
\]
and (ii) with updated \(T_r\) fixed, perform a steepest-descent update for \(T_c\) in the direction of 
\[
Z_c = T_c (D_T \cdot \bar{J}_i(T_r)) \cdot T_r - D_T \cdot \bar{J}_i(T_r)
\]
The use of Armijo step-size in each update ensures that \(J_i(T)\) can never increase, and hence such alternating updates on \(O(L)\) are guaranteed to produce a sequence of \(T = T_r \otimes T_c\) on \(O(K)\) for which \(J_i(T)\) always converges at least to a local minimum.
(PSNR), typically computed over an image or a video frame. In order to better capture the gains due to adaptive coding, we will use the average local PSNR which is the average of the PSNRs of all SBs.

When applying the Algorithm 2 to image data, an important consideration is the choice of the initial codebook of \( N \) transform matrices each of which must be orthonormal. One simple approach is to estimate the KLT matrix of each SB in the training set and randomly choose \( N \) matrices from this set. We also considered an alternative method wherein the training set is first classified into \( N \) subsets based on the DCT energy spectrum of SBs, and the KLT of each subset was used to form the initial codebook (rationale is that the transform that produces the best energy compaction must be nearly the same for those SBs with similar energy spectra). It was found that the latter method generally produced better final codebooks, particularly when the codebook size is large.

A. Still images

Applying adaptive transform coding to images would require the segmentation of images into SBs of pixels. In our experiments, we divided images into a fixed-size 32 × 32 non-overlapping blocks to form SBs. The DC component of each SB was then subtracted and quantized separately. Next, each DC-removed SB was sub-divided into 8 × 8 non-overlapping blocks, referred to as transform blocks (TBs) all of which were to be transform coded using a common transform matrix chosen from a pre-designed codebook. We considered both non-separable 64 × 64 and separable 8 × 8 transforms. We designed transform matrix codebooks using training set of 101 natural images which include 512 × 512 and 3072 × 2048 gray-scale images with 8-bits/pixel resolution. For each DC-removed SB, the 64 × 64 covariance matrix of the constituent TBs was estimated. Since the number of TBs per SB available to estimate the full covariance matrix was small, we constrained the structure of the covariance matrix by assuming the correlation function of the pixels to be spatially invariant inside each SB. As the 2D-DCT would be very effective for SBs resembling a Gaussian process and there could be many such SBs in a given image, we assumed the 2D-DCT to be one of the possible transforms and optimized the rest of the codebook for SBs in the training set which were unlikely to be coded well by the 2D-DCT. To this end, we used the energy compaction (EC) of the 2D-DCT relative to the KLT of each SB as the criterion to prune the initial training set, where the EC was measured by the percentage of energy in the first 8 transform coefficients out of the total of 64. The training set was pruned by eliminating SBs for which EC(DCT)/EC(KLT) > 65%. Note that when the 2D-DCT was included in a non-separable transform codebook, the Kronecker-product of the row and column DCT matrices was used, see (12).

As in the case of the simple Gaussian-mixture example in Sec. III-D, we observed that a codebook optimized for an appropriately chosen single \( \Delta \) value can be nearly optimal for a range of bit rates. In our image coding experiments, we used rates below 1 bits per pixel (bpp) for which we found codebooks designed for \( \Delta = 55 \) to be as nearly as good as
the optimal design for each bit rate.

It has been shown in past work that both image and video residual transform coefficients are well modeled by Laplace pdf than the Gaussian, see for example [9], [10]. In all our experiments with image and video residuals we observed that the codebooks optimized with the finite-rate Laplace model (Sec. III-B2) were always better than the high-rate Gaussian model (Sec. III-B1). For example, Fig. 3 compares the PSNR observed in adaptive coding of the Barbara image using size $N = 9$ codebooks optimized with the two quantizer MSE models. All experimental results presented in this section, as well as in Sec. V-B, have been obtained with the finite-rate Laplacian model.

In order to investigate the potential benefits of using an optimized codebook over the standard 2-DCT, we designed both non-separable and separable codebooks of different sizes for many different rates in the range of 0.1 - 1 bpp. We tested the codebooks on many standard and non-standard images and observed that the adaptive coding gain very much depended on the image. As one would expect, the local regions in images that resembled Gaussian noise did not benefit much from adaptive coding. Furthermore, as the 2D-DCT uses sinusoidal basis functions in horizontal and vertical directions, for image regions with textures oriented in these directions, adaptive transform coding may not yield significant improvements. We applied adaptive coding (using the codebooks trained for natural images as described above) to the computer art image Retoka Big Data (http://retoka.com/big-data) shown in Fig. 4(a), which has textures in vertical, horizontal, as well as diagonal orientations. Fig. 4(b) shows the SBs for which a transform matrix other than the 2D-DCT was picked (coded here with a $N = 17$, 64×64 non-separable transform codebook and a $\Delta = 65$), which clearly demonstrates the sub-optimality of the 2D-DCT for diagonal textures. Tables I (non-separable backwards 64×64 transforms) and II (separable 8×8 transforms) compare the PSNR gains (over the 2D-DCT) achieved by the proposed QOT codebooks and the alternative SOT codebooks [17] for several codebooks sizes and bit rates. Also shown in these tables for each case is the ABRO as a percentage of the total bit rate. Our image coding experiments involving many images suggested that this penalty was in the range of 0.15-

| $N$ | Rate | % ABRO | PSNR gain (dB) |
|-----|------|--------|----------------|
|     | QOT  | SOT    | QOT  | SOT    |
| 5   | 0.25 | 0.58   | 0.77 | 1.81   | 1.51 |
|     | 0.41 | 0.36   | 0.40 | 2.49   | 2.20 |
|     | 0.63 | 0.22   | 0.24 | 2.79   | 2.28 |
| 9   | 0.25 | 0.68   | 0.74 | 1.41   | 1.39 |
|     | 0.41 | 0.38   | 0.36 | 2.60   | 1.96 |
|     | 0.63 | 0.22   | 0.21 | 2.96   | 2.32 |
| 17  | 0.25 | 0.73   | 0.94 | 2.02   | 1.02 |
|     | 0.41 | 0.40   | 0.48 | 3.46   | 1.38 |
|     | 0.63 | 0.22   | 0.17 | 4.26   | 1.42 |
| 33  | 0.25 | 0.83   | 0.85 | 2.46   | 1.88 |
|     | 0.41 | 0.40   | 0.47 | 4.08   | 2.62 |
|     | 0.63 | 0.22   | 0.23 | 4.75   | 2.27 |
| 65  | 0.25 | 1.00   | 0.75 | 2.48   | 1.94 |
|     | 0.41 | 0.40   | 0.38 | 4.11   | 2.90 |
|     | 0.63 | 0.22   | 0.23 | 4.96   | 2.35 |

Fig. 5. Adaptive coding of 512×512 Barbara image using a transform codebook of size 17 in which one of the codewords is the 2D-DCT. The squares with a cross mark 32×32 locally stationary blocks that have been adaptively coded using a transform other than the 2D-DCT. Quantization step size is $\Delta = 93$.  

![Image](http://retoka.com/big-data)
TABLE III

| N   | Rate (bpp) | % ABRO | PSNR gain (dB) |
|-----|------------|--------|----------------|
|     |            | QOT    | SOI            | QOT | SOI |
| 9   | 0.17       | 0.72   | 0.68           | 0.10 | -0.04 |
|     | 0.40       | 0.28   | 0.20           | 0.15 | 0.05  |
|     | 0.62       | 0.16   | 0.08           | 0.11 | 0.03  |
| 17  | 0.17       | 1.13   | 0.78           | 0.18 | 0.01  |
|     | 0.40       | 0.34   | 0.16           | 0.26 | 0.01  |
|     | 0.62       | 0.21   | 0.06           | 0.19 | -0.01 |
| 33  | 0.4        | 0.39   | 0.19           | 0.33 | 0.09  |
|     | 0.62       | 0.25   | 0.07           | 0.33 | 0.04  |
| 65  | 0.17       | 1.83   | 0.64           | 0.27 | 0.00  |
|     | 0.40       | 0.48   | 0.14           | 0.44 | 0.01  |
|     | 0.62       | 0.27   | 0.03           | 0.37 | -0.01 |

TABLE IV

| N   | Rate (bpp) | % ABRO | PSNR gain (dB) |
|-----|------------|--------|----------------|
|     |            | QOT    | SOI            | QOT | SOI |
| 9   | 0.17       | 0.98   | 1.11           | 0.06 | -0.02 |
|     | 0.4        | 0.39   | 0.35           | 0.10 | 0.05  |
|     | 0.62       | 0.23   | 0.24           | 0.08 | 0.04  |
| 17  | 0.17       | 1.10   | 1.40           | 0.06 | -0.02 |
|     | 0.4        | 0.45   | 0.52           | 0.10 | -0.06 |
|     | 0.62       | 0.30   | 0.32           | 0.02 | -0.12 |
| 33  | 0.4        | 0.61   | 0.58           | 0.16 | -0.04 |
|     | 0.62       | 0.38   | 0.36           | 0.07 | -0.06 |
| 65  | 0.17       | 1.99   | 1.59           | 0.11 | -0.01 |
|     | 0.4        | 0.76   | 0.62           | 0.17 | -0.11 |
|     | 0.62       | 0.49   | 0.41           | 0.10 | -0.15 |

TABLE V

| N   | Rate Drop (%) | PSNR Gain (dB) |
|-----|---------------|----------------|
|     | without ABRO  | with ABRO      |
| 5   | 3.16          | 2.01           | 0.15          | 0.22 |
| 17  | 3.74          | 1.81           | 0.20          | 0.25 |
| 65  | 4.08          | 1.40           | 0.25          | 0.28 |
| 257 | 4.20          | 0.99           | 0.26          | 0.29 |
| 1025| 4.36          | 0.88           | 0.24          | 0.28 |

An example of a natural image which contains regions with “stripe-textures” is the Barbara image shown in Fig. 5. This figure shows the SBs for which a codeword other than the 2D-DCT was used as the transform in adaptive coding (recall that figure shows the SBs for which a codeword other than the 2D-DCT was used as the transform in adaptive coding (recall that the 2D-DCT is also in the codebook), along with examples of 32 × 32 SBs which are clearly better coded by QOTs than the 2D-DCT. Tables III and IV (64 × 64 non-separable transforms) and V (8 × 8 separable transforms) compare the PSNR gains achieved by QOT and SOT codebooks for this image. Note that to achieve the same bit rate, different quantization step-sizes must be used with the 2D-DCT-based non-adaptive coding, and adaptive coding with QOT and SOT, due to the differences in energy compaction as well as the ABRO in adaptive coding. While the gains of adaptive coding for this natural image are not as significant as in the case of the Big Data image, QOTs achieve a noticeably better gain than the SOTs. The PSNR gains of SOTs in this case are not large enough to offset the loss due to ABRO.

B. MCPE images

We applied adaptive transform coding to MCPE image sequences extracted from an H265 standard video encoder [30]. Our source video included 9 widely used standard CIF resolution (352 × 288), 30 fps gray-scale video sequences (Bus, Coast guard, Crew, Football, Foreman, Mobile, Soccer, Stefan, and Tennis). Each video was encoded using a group of picture (GOP) size of 8 frames, and except for the first frame in a sequence, all frames were coded as prediction (P) frames. For the purpose of adaptive transform coding, we defined spatial SBs by dividing each MCPE frame into 32×32 non-overlapping blocks. Each spatial SB was sub-divided into 8×8-pixel to create TBs (64-dimensional vectors.) As with the still images, one could encode all TBs in a given spatial SB using a common transform. However, as consecutive MCPE frames are often correlated, a further coding gain may be achieved by grouping time-aligned 32×32 spatial SBs in several consecutive frames to form spatio-temporal SBs, and encoding all TBs in each spatio-temporal SB using a single transform. A gain is possible because of the reduction of the ABRO. In our experiments, we have considered spatio-temporal SBs of size 32×32×8 as well as spatial-only SBs of size 32×32. As in the case of image coding, we included the 2D-DCT as a possible transform in every codebook and trained the rest of the codebook using a pruned training set. To this end, only SBs with ∆2D-DCT < 60% were selected to train the codebooks. Experimental results presented in this section are for rates less than 1 bpp, and for this rate range, the codebooks designed for ∆ = 45 were found to be nearly optimal. Note that the experimental results presented in this section reflect the PSNRs and bit rates (estimated quantization index entropies) of transform coding the MCPE image sequence only (ignoring other meta-data produced by the video codec, such as the motion vectors.)

Tables V (Akiyo) and VI (Mother and Daughter) show the PSNR gains and rate reductions (Rate Drop) achieved by
64 × 64 non-separable QOT codebooks for MCPE images when the quantization step-size \( \Delta = 18 \) and SBs are of size 32 × 32 is used. At the same \( \Delta \), QOT codebooks produce better energy compaction compared to the non-adaptive 2D-DCT and hence a PSNR gain and a rate drop. However, the effective rate advantage is significantly degraded by the ABRO when the SB size is smaller. The tables show both the overall PSNR gains (considering all SBs) and the PSNR gains of only those SBs for which a transform other than the 2D-DCT have been selected (since the quantization step-size is the same in adaptive and non-adaptive cases, a PSNR gain is achieved only in non-DCT coded SBs.)

### Tables

#### Table VI
**PSNR Gain of Adaptive Coding with 64 × 64 Non-separable Transform Codebooks for MCPE Image Sequence of Mother-Daughter Video Sequence.** PSNR of non-adaptive coding with 2D-DCT is 39.9 dB and 43.1 dB at rates 500 kbps and 1380 kbps respectively.

| \( N \) | Rate Drop (%) | PSNR Gain (dB) |
|---|---|---|
| | without ABRO | with ABRO | Overall | non-DCT SBs only |
| 5 | 1.63 | 0.62 | 0.15 | 0.22 |
| 17 | 1.85 | 0.05 | 0.21 | 0.25 |
| 65 | 2.21 | -0.16 | 0.26 | 0.29 |
| 257 | 2.61 | -0.42 | 0.27 | 0.30 |
| 1025 | 2.92 | -0.35 | 0.27 | 0.30 |

#### Table VII
**PSNR Gain of Adaptive Coding with 64 × 64 Non-separable Transform Codebooks for MCPE Image Sequence of Mother-Daughter Video Sequence.** PSNR of non-adaptive coding with 2D-DCT is 39.9 dB and 43.1 dB at rates 500 kbps and 1380 kbps respectively.

| \( N \) | Rate (kbps) | % non-DCT SBs | PSNR Gain (dB) |
|---|---|---|---|
| | QOT | SOT | QOT | SOT |
| 5 | 500 | 75 | 40 | 0.2 | 0.1 |
| 17 | 500 | 84 | 54 | 0.2 | 0.2 |
| 65 | 500 | 88 | 63 | 0.3 | 0.2 |
| 257 | 500 | 87 | 72 | 0.3 | 0.2 |
| 1025 | 500 | 86 | 66 | 0.4 | 0.3 |

#### Table VIII
**PSNR Gain of Adaptive Coding with 64 × 64 Non-separable Transform Codebooks for MCPE Image Sequence of Akiyo Video Sequence.** PSNR of non-adaptive coding with 2D-DCT is 41.3 dB and 43.8 dB at rates 500 kbps and 1165 kbps respectively.

| \( N \) | Rate (kbps) | % non-DCT SBs | PSNR Gain (dB) |
|---|---|---|---|
| | QOT | SOT | QOT | SOT |
| 5 | 500 | 69 | 45 | 0.2 | 0.1 |
| 17 | 500 | 82 | 54 | 0.2 | 0.2 |
| 65 | 500 | 90 | 62 | 0.2 | 0.2 |
| 257 | 500 | 87 | 67 | 0.2 | 0.2 |
| 1025 | 500 | 83 | 67 | 0.2 | 0.2 |

#### Table IX
**PSNR Gain of Adaptive Coding with 8 × 8 Separable Transform Codebooks for MCPE Image Sequence of Mother-Daughter Video Sequence.** PSNR of non-adaptive coding with 2D-DCT is 39.9 dB and 43.1 dB at rates 500 kbps and 1380 kbps respectively.

| \( N \) | Rate (kbps) | % non-DCT SBs | PSNR Gain (dB) |
|---|---|---|---|
| | QOT | SOT | QOT | SOT |
| 5 | 500 | 83 | 50 | 0.2 | 0.1 |
| 17 | 500 | 93 | 62 | 0.3 | 0.2 |
| 65 | 500 | 96 | 82 | 0.3 | 0.3 |
| 257 | 500 | 97 | 91 | 0.4 | 0.4 |
| 1025 | 500 | 98 | 94 | 0.5 | 0.4 |

#### Table X
**PSNR Gain of Adaptive Coding with 8 × 8 Separable Transform Codebooks for MCPE Image Sequence of Akiyo Video Sequence.** PSNR of non-adaptive coding with 2D-DCT is 41.3 dB and 43.8 dB at rates 500 kbps and 1165 kbps respectively.

| \( N \) | Rate (kbps) | % non-DCT SBs | PSNR Gain (dB) |
|---|---|---|---|
| | QOT | SOT | QOT | SOT |
| 5 | 500 | 79 | 48 | 0.2 | 0.1 |
| 17 | 500 | 90 | 62 | 0.3 | 0.2 |
| 65 | 500 | 99 | 83 | 0.3 | 0.3 |
| 257 | 500 | 98 | 91 | 0.4 | 0.3 |
| 1025 | 500 | 99 | 94 | 0.4 | 0.4 |

Tables VII (Mother sequence) and VIII (Akiyo sequence) compare the performance of adaptive coding using 64 × 64 non-separable QOT and SOT codebooks for spatio-temporal SBs of size 32 × 32 × 8. Also shown in these tables are the % of SBs coded by a transform other than the 2D-DCT (2D-DCT is one of the codewords in all codebooks.) Similar results for 8 × 8 separable transforms are shown in Tables IX and X. In our adaptive coding experiments, it was generally observed that the SBs in MCPE images were more often coded with transforms that were different to the 2D-DCT, than the SBs in still images, mainly due to the fact that video prediction errors have characteristics very different from natural images. Note also that, for MCPE images separable transform codebooks yield higher PSNR gains compared to the non-separable counterparts. Even though in principle the best separable transform cannot outperform the best non-separable transform, this is not necessarily the case with transforms estimated from a limited amount of data. When estimated using the same amount of data, a non-separable transform suffers from a higher estimation error, as it contains a larger number of free parameters than a separable transform of the same size. Separable transforms can already be highly effective for MCPE images which often have more prominent directional characteristics (due to high prediction errors around moving objects) than in natural images.
VI. Conclusion

We proposed a general data-driven algorithm to design orthonormal matrix codebooks for adaptive transform coding. This algorithm can be used to design transforms for any non-stationary vector process which can be considered locally stationary, with the only requirement being the availability of a simple probability model for the transform coefficients. We focused on Gaussian and Laplacian distributions which have wide applicability in image and video compression. Even though the optimal transform can be bit rate dependent when the source is non-Gaussian, our experiments with natural images and video sequences showed that a single codebook of transforms optimized for a carefully chosen quantization step size can be nearly optimal for a wide range of bit rates. We compared codebooks designed using the proposed algorithm, which is based on directly minimizing the transform coding MSE, with those designed with the sparse orthonormal transform method [17]. Even though the two methods are very different, our experimental results showed that both produce transform codebooks with nearly identical coding performance for natural images and video.

The PSNR improvements of adaptive transform coding over non-adaptive coding in natural images and videos that we have reported here are modest. This is likely due to the fact that we have used fixed-size blocks on a regular grid as locally stationary blocks. Since non-stationarities in natural images and video sequences showed that a single codebook for natural images and video.

APPENDIX

A. Derivatives for steepest descent: non-separable transforms (Laplacian model)

Let \( \theta_k(T, C_B, \Delta) = g(b_k(T)) \) where \( b_k(T) = \sqrt{\frac{2}{\pi \gamma_k(T) \Delta}} \). Using the chain rule, we can write

\[
D_T [\theta_k(T, C_B, \Delta)] = \frac{d}{db_k} g(b_k(T)) D_T [b_k].
\]

It is easy to show that

\[
\alpha(b_k) = 4b_k - e^{-\frac{\Delta^2}{2b_k^2}} \left( 1 + \frac{z^2 - 2\Delta}{4} + z b_k + \Delta b_k \frac{e^\Delta - 1}{e^\Delta - 1} \right) - e^{-\frac{\Delta^2}{2b_k^2}} \left( 1 + \frac{\Delta b_k e^{-\frac{\Delta^2}{2b_k^2}} - \Delta b_k + 2\Delta^2 e^{-\frac{\Delta^2}{2b_k^2}}}{b_k (e^\Delta - 1)^2} \right),
\]

and \( D_T [b_k] = \frac{e_k e_k^T T C_B}{2b_k} \).

Next, let \( r_k(T, C_B, \Delta) = h(b_k(T)) \). Using the chain rule

\[
D_T [r_k(T, C_B, \Delta)] = \frac{d}{db_k} h(b_k(T)) D_T [b_k].
\]

It can be shown that

\[
\frac{d}{db_k} h(b_k) = \gamma_{k,1} - \gamma_{k,2} + \gamma_{k,3},
\]

where

\[
\gamma_{k,1} = \frac{z \mu_k}{2b_k^2 \ln(2)} (\ln(1 - \mu_k) + 1),
\]

\[
\gamma_{k,2} = -\frac{\mu_k}{2b_k^2 \ln(2)} \left( z \ln \left( \frac{\mu_k \eta_k}{\eta_k} \right) + \frac{z \eta_k - 2\Delta \zeta_k}{\eta_k} \right),
\]

\[
\gamma_{k,3} = \Delta \mu_k \zeta_k \frac{z \Delta - 2b_k \Delta - z + 2\Delta}{2b_k^2 (\zeta_k + 1)^2 \ln(2)},
\]

\[
\mu_k = e^{-\frac{\Delta^2}{2b_k^2}}, \zeta_k = e^{\frac{\Delta}{b_k}}, \text{ and } \eta_k = e^{\frac{\Delta}{b_k}} - 1. \text{ Therefore,}
\]

\[
D_T [r_k(T, C_B, \Delta)] = \left( \gamma_{k,1} - \gamma_{k,2} + \gamma_{k,3} \right) \frac{e_k e_k^T T C_B}{2b_k}. \tag{15}
\]

B. Derivatives for steepest decent: separable transforms

First we define the \( K \times 1 \) vector

\[
q_k(T) = (T_r \otimes T_c)^T e_k = vec(T_r^T E_k T_r),
\]

where \( L \times L \) matrix \( E_k = vec^{-1}(e_k) \). Then, \( \Phi \) can be written as

\[
\sigma^2_{Y_k}(T) = [q_k(T)]^T C_B q_k(T). \tag{17}
\]

1) High-rate Gaussian model: First note that for fixed \( \Delta \), \( D_{T_r} [\theta_k(T, C_B, \Delta)] = 0 \) and \( D_{T_c} [\theta_k(T, C_B, \Delta)] = 0 \). Next, using \( \Phi \), \( \Phi \), and the chain rule we have

\[
\frac{d}{dq_k} \frac{r_k(T, C_B, \Delta)}{d (vec(T_c))} = \frac{d}{dq_k} \frac{r_k(T, C_B, \Delta)}{d (vec(T_r))} \frac{dq_k(T)}{d (vec(T_c))},
\]

where

\[
\frac{d}{dq_k} \frac{r_k(T, C_B, \Delta)}{d (vec(T_c))} = \frac{1}{\ln(2) \sigma^2_{Y_k}(T)} [q_k(T)]^T C_B,
\]

is a \( K \times K \) derivative matrix. Note that we have used the fact that \( vec(ABC) = (C^T \otimes A)vec(B) \). We can thus state \( \Phi \) in the matrix form as

\[
D_{T_c} [r_k(T, C_B, \Delta)] = \frac{F_k(T)}{\ln(2) \sigma^2_{Y_k}(T)}, \tag{19}
\]

where \( F_k(T) = vec^{-1} \left( [q_k(T)]^T C_B (I_L \otimes (T_c^T E_k))^T \right) \).

Next consider the \( 1 \times K \) derivative vector

\[
\frac{d}{dq_k} \frac{r_k(T, C_B, \Delta)}{d (vec(T_c))} = \frac{d}{dq_k} \frac{r_k(T, C_B, \Delta)}{d (vec(T_c))} \frac{dq_k(T)}{d (vec(T_c))}, \tag{20}
\]
where
\[
\frac{dq_k(T)}{d(\text{vec}(T_c))} = \frac{d}{d(\text{vec}(T_c))} \text{vec}(T_c^T E_k^TE_r),
\]
\[
= \frac{d}{d(\text{vec}(T_c))} ((E_k^TE_r)^T \otimes I_L) \text{vec}(T_c^T),
\]
\[
= \frac{d}{d(\text{vec}(T_c))} ((E_k^TE_r)^T \otimes I_L) K_K \text{vec}(T_c),
\]
where $K_K$ is the commutation matrix. We can thus state (20) in matrix form as
\[
D_{T_c} [r_k(T, C_B, \Delta)] = G_k(T),
\]
where $G_k(T) = \text{vec}^{-1} \left( (q_k(T))^T C_B ((E_k^TE_r)^T \otimes I_L) K_K \right)^T$.

2) Finite-rate Laplacian model: Using (9) and (16), we have
\[
b_k(T) = \sqrt{\frac{\sigma^2_k(T)}{2}} = \sqrt{\frac{(q_k(T))^T C_B q_k(T)}{2}}.
\]

Using the chain-rule, we can write the $1 \times K$ derivative vectors
\[
\frac{d\theta_k(T, C_B, \Delta)}{d(\text{vec}(T_c))} = \frac{d\theta_k(T, C_B, \Delta)}{d\text{vec}(T_c)} = \frac{d\theta_k(T, C_B, \Delta)}{d\text{vec}(T_c)} = \frac{d\theta_k(T, C_B, \Delta)}{d\text{vec}(T_c)} = \frac{d\theta_k(T, C_B, \Delta)}{d\text{vec}(T_c)}.
\]

We can state (22) and (23) respectively in matrix form as
\[
D_{T_c} [\theta_k(T, C_B, \Delta)] = \alpha (b_k(T)) F_k(T),
\]
\[
D_{T_c} [\theta_k(T, C_B, \Delta)] = \alpha (b_k(T)) G_k(T),
\]
We can again use the chain rule to write
\[
\frac{dr(T, C_B, \Delta)}{d(\text{vec}(T_c))} = \frac{dr(T, C_B, \Delta)}{d\text{vec}(T_c)} = \frac{dr(T, C_B, \Delta)}{d\text{vec}(T_c)} = \frac{dr(T, C_B, \Delta)}{d\text{vec}(T_c)} = \frac{dr(T, C_B, \Delta)}{d\text{vec}(T_c)}.
\]

We can state (26) and (27) respectively in matrix form as
\[
D_{T_c} [\gamma_k(T, C_B, \Delta)] = (\gamma_{k,1} - \gamma_{k,2} + \gamma_{k,3}) F_k(T),
\]
\[
D_{T_c} [\gamma_k(T, C_B, \Delta)] = (\gamma_{k,1} - \gamma_{k,2} + \gamma_{k,3}) G_k(T),
\]
where $T = E_r \otimes T_e$.

REFERENCES

[1] A. Gersho and R. M. Gray, Vector quantization and signal compression. Norwell, MA, USA: Kluwer Academic Publishers, 1991.
[2] D. S. Taubman and M. W. Marcellin, JPEG2000: Image Compression Fundamentals, Standards and Practice. Kluwer Academic Publishers, 2002.
[3] T. Nguyen, P. Helle, M. Winken, B. Bross, D. Marpe, H. Schwarz, and T. Wiegand, “Transform coding techniques in HEVC,” IEEE J. Sel. Topics Signal Process., vol. 7, no. 6, pp. 978–989, Aug. 2013.
[4] V. K. Goyal, J. Zhuang, and M. Vetterli, “Transform coding with backward adaptive updates,” IEEE Trans. Inf. Theory, vol. 46, no. 4, pp. 1623–1633, Jul. 2000.
[5] M. Effros, H. Feng, and K. Zeger, “Suboptimality of the Karhunen-Loève transform for transform coding,” IEEE Trans. Inf. Theory, vol. 50, no. 8, pp. 1605–1619, Aug. 2004.
[6] P. Boragolla and P. Yahampath, “Orthornomal matrix codebook design for adaptive transform coding,” in Proc. IEEE DCC, Mar. 2002, p. 442.
[7] D. P. Bertsekas, Nonlinear Programming, 2nd ed. Belmont, MA: Athena Scientific, 1999.
[8] J. Sun, Y. Duan, J. Li, J. Liu, and Z. Guo, “Rate-distortion analysis of dead-zone plus uniform threshold scalar quantization and its application—Part I: Fundamental theory,” IEEE Trans. Image Process., vol. 29, no. 1, pp. 202–214, Jan. 2013.
[9] E. Lam and J. Goodman, “A mathematical analysis of the DCT coefficient distributions for images,” IEEE Trans. Image Process., vol. 9, no. 10, pp. 1661–1666, 2000.
[10] X. Li, N. Oertel, A. Hutter, and A. Kaup, “Laplace distribution based Lagrangian rate distortion optimization for hybrid video coding,” IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 2, pp. 195–205, Feb. 2009.
[11] H. Schwarz, M. Coban, M. Karczewicz, T.-D. C. anbd F. Bossen, A. Alshin, J. Lainema, and C. R. Helmrich, “Quantization and entropy coding in the versatile video coding (VVC) standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 31, pp. 3891–3906, Apr. 2021.
[12] J. H. Manton, “Geometry, manifolds, and nonconvex optimization,” IEEE Signal Process. Mag., vol. 37, no. 5, pp. 1605–1619, Sep. 2020.
[13] Z. Gu, W. Lin, B. B. Lee, and C. T. Lau, “Rotated orthogonal transform (ROT) for motion-compensation residual coding,” IEEE Trans. Image Process., vol. 21, no. 12, pp. 4770–4781, Dec. 2012.
[14] H. Chen and B. Zeng, “New transforms tightly bounded by DCT and KLT,” IEEE Signal Process. Lett., vol. 19, no. 6, pp. 344–347, Apr. 2012.
[15] M. A. Sadrahbadi, A. K. Khandani, and F. Lahouti, “A new method of channel feedback quantization for high data rate MIMO systems,” in IEEE Globecom, 2004, pp. 91–95.
[16] R. Shepard, S. R. Brozell, and G. Gidofalvi, “The representation and parametrization of orthogonal matrices,” J. Phys. Chem. A, vol. 119, no. 28, pp. 7924–793, May 2015.
[17] O. G. Sezer, O. G. Guleryuz, and Y. Alunbas, “Approximation and compression with sparse orthonormal transforms,” IEEE Trans. Image Process., vol. 24, no. 8, pp. 2328–2343, Aug. 2015.
[18] C. Chang and B. Girod, “Direction-adaptive discrete wavelet transform for image compression,” IEEE Trans. Image Process., vol. 16, no. 5, pp. 1289–1302, May 2007.
[19] B. Zeng and J. Fu, “Directional discrete cosine transforms—A new framework for image coding,” IEEE Trans. Circuits Syst. Video Technol., vol. 18, no. 3, pp. 305–3013, Mar. 2008.
[20] J.-W. Kang, M. Gabbouj, and C.-C. J. Kuo, “Sparse/DCT (S/DCT) two-layered representation of prediction residuals for video coding,” IEEE Trans. Image Process., vol. 22, no. 7, pp. 2711–2722, Jul. 2013.
[21] F. Kamisli and J. S. Lim, “1-D transforms for the motion compensation residual,” IEEE Trans. Image Process., vol. 20, no. 4, pp. 1036–1046, Oct. 2010.
[22] F. Zou, O. C. Au, C. Pang, J. Dai, X. Zhang, and L. Fang, “Rate-distortion optimized transforms based on the Lloyd-type algorithm for intra block coding,” IEEE J. Sel. Topics Signal Process., vol. 7, no. 6, pp. 1072 – 1083, Dec. 2013.
[23] X. Zhao, L. Zhang, S. Ma, and W. Gao, “Video coding with rate-distortion optimized transform,” IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 1, pp. 138–151, May 2011.
[24] X. Zhao, J. Chen, M. Karczewicz, A. Said, and V. Seregin, “Joint separable and non-separable transforms for next-generation video coding,” IEEE Trans. Image Process., vol. 27, no. 5, pp. 2514–2525, May 2018.
[25] M. Effros, P. A. Chou, and R. M. Gray, “Weighted universal image compression,” IEEE Trans. Image Process., vol. 8, no. 10, pp. 1317–1329, Oct. 1999.
[26] C. Archer and T. K. Leen, “Adaptive transform coding as constrained vector quantization,” in Proc. Neural Networks for Sig. Proc. X, vol. 1, 2000, pp. 308–317.
[27] G. Martinielli, L. P. Ricotti, and G. Marcone, “Neural clustering for optimal KLT image compression,” IEEE Signal Process. Lett., vol. 41, no. 4, pp. 1737–1739, Apr. 1993.
[28] R. D. Dony and S. Haykin, “Optimally adaptive transform coding,” IEEE Trans. Image Process., vol. 4, no. 10, pp. 1358–1370, Oct. 1995.
[29] H. Caglar, S. Güntürk, B. Sankur, and E. Anarim, “VQ-adaptive block
transform coding of images,” *IEEE Trans. Image Process.*, vol. 7, no. 1,
pp. 110–115, Jan. 1998.
[30] C. Archer and T. K. Leen, “A generalized Lloyd-type algorithm for
adaptive transform coder design,” *IEEE Trans. Signal Process.*, vol. 52,
no. 1, pp. 255–264, Jan. 2004.
[31] S. Ma, W. Gao, and Y. Lu, “Rate-distortion analysis for H.264/AVC
video coding and its application to rate control,” *IEEE Trans. Circuits
Syst. Video Technol.*, vol. 15, no. 12, pp. 1533–1544, Dec. 2005.
[32] J. H. Manton, “Optimization algorithms exploiting unitary constraints,”
*IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 635–650, Aug. 2002.
[33] H.-M. Hang and J.-J. Chen, “Source model for transform video coder
and its application — Part I: Fundamental theory,” *IEEE Trans. Circuits
Syst. Video Technol.*, vol. 7, no. 2, pp. 287–298, Apr. 1997.
[34] J. Cui, S. Wang, S. Wang, X. Zhang, S. Ma, and W. Gao, “Hybrid
Laplace distribution-based low complexity rate-distortion optimized
quantization,” *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3802–
3816, 2017.
[35] Y.-K. Tu, J.-F. Yang, and M.-T. Sun, “Rate-distortion modeling for effi-
cient H.264/AVC encoding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 5, pp. 530–543, 2007.
[36] T. M. Cover and J. Thomas, *Elements of information theory.* John
Wiley & Sons, 1999.
[37] J. R. Magnus and H. Neudecker, *Matrix differential calculus with
applications in statistics and econometrics.* John Wiley & Sons, 2007.
[38] A. K. Jain, *Fundamentals of Digital Image Processing.* Prentice Hall,
1989.
[39] R. Strickland, “Estimation of local statistics for digital processing of
nonstationary images,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 2, pp. 465–469, 1985.
[40] “HEVC reference software (HVEC test model),” Joint Collaborative
Team on Video Coding (JCT-VC), https://hevc.hhi.fraunhofer.de.