# Optimized Signal Expansions for Sparse Representation

Sven Ole Aase, John Håkon Husøy, Karl Skretting, and Kjersti Engan

***Abstract*—Traditional signal decompositions such as transforms, filterbanks, and wavelets generate signal expansions using the *analysis–synthesis* setting: The expansion coefficients are found by taking the inner product of the signal with the corresponding analysis vector. In this paper, we try to free ourselves from the analysis–synthesis paradigm by concentrating on the *synthesis* or *reconstruction* part of the signal expansion. Ignoring the analysis issue completely, we construct sets of synthesis vectors, which are denoted *waveform dictionaries*, for efficient signal representation. Within this framework, we present an algorithm for designing waveform dictionaries that allow *sparse* representations: The objective is to approximate a training signal using a small number of dictionary vectors. Our algorithm optimizes the dictionary vectors with respect to the average nonlinear approximation error, i.e., the error resulting when keeping a fixed number $n$ of expansion coefficients but not necessarily the *first* $n$ coefficients. Using signals from a Gaussian, autoregressive process with correlation factor 0.95, it is demonstrated that for established signal expansions like the Karhunen–Loéve transform, the lapped orthogonal transform, and the biorthogonal 7/9 wavelet, it is possible to improve the approximation capabilities by up to 30% by fine tuning of the expansion vectors.**

## I. Introduction

A *signal expansion* is simply a weighted sum of vectors $\mathbf{f}_i$. This weighted sum may be identical to, or an approximation to, a given signal vector $\mathbf{x}$. If the expansion is identical to $\mathbf{x}$, we can write

$$\mathbf{x} = \mathcal{F}\mathbf{w} = \sum_i w_i \mathbf{f}_i \tag{1}$$

where $\mathcal{F}$ is a (possibly infinite) matrix with $\{\mathbf{f}_i\}$ as columns, and $\mathbf{w}$ is the vector of expansion coefficients. Equation (1) can be interpreted as a *synthesis* formula in the sense that $\mathbf{x}$ is synthesized, or build up, from a library of expansion vectors using appropriately selected values for the expansion coefficients. For this reason, $\mathcal{F}$ is sometimes referred to as a *waveform dictionary.* If the matrix $\mathcal{F}$ is invertible, a unique set of coefficients for the exact representation of any signal vector $\mathbf{x}$ can be obtained as[1]

$$\mathbf{w} = \mathcal{F}^{-1}\mathbf{x} \tag{2}$$

Manuscript received September 24, 1999; revised January 9, 2001. The associate editor coordinating the review of this paper and approving it for publication was Prof. Lang Tong.

The authors are with Høgskolen i Stavanger, Department of Electrical and Computer Engineering, Stavanger, Norway (e-mail: Sven.O.Aase@tn.his.no).

Publisher Item Identifier S 1053-587X(01)03338-4.

[1]For notational convenience, we denote the *forward* matrix by $\mathcal{F}^{-1}$ and the *inverse* or *reconstruction* matrix by $\mathcal{F}$.

and this is commonly referred to as the *analysis* equation in an analysis–synthesis setting. Note that in the interest of maximum generality, we have not specified the dimensions of the matrices and vectors involved in (1) and (2). Depending on the dimensions, which may extend to infinity, as well as the structure of the $\mathcal{F}$ matrix, the analysis–synthesis equations given above cover many important cases including transforms, filterbanks, wavelets, and wavelet packets.

The main objective for using the analysis–synthesis framework in signal processing applications is to construct $\mathcal{F}$ such that the vector of coefficients $\mathbf{w}$ is more attractive to work with than $\mathbf{x}$. For signal representation purposes, a crucial point is that $\mathbf{w}$ should be *sparse*. The sparseness constraint refers to the requirement that $\mathbf{w}$ must have as few nonzero elements as possible [1]. In a data compression setup, the sparseness constraint facilitates bit-efficient representation of the original vector $\mathbf{x}$ since only the nonzero elements of $\mathbf{w}$ have to be quantized and stored or transmitted.

In the present work, we aim at freeing ourselves from the traditional analysis–synthesis paradigm in that we concentrate on the *synthesis* or *reconstruction* part of the signal expansion. That is, given the coefficients $\{w_i\}$, the reconstructed signal vector $\tilde{\mathbf{x}}$ is given by

$$\tilde{\mathbf{x}} = \mathcal{F}\mathbf{w} = \sum_i w_i \mathbf{f}_i. \tag{3}$$

The consequences of ignoring the analysis part of the expansion can be summarized as follows.

1) There are no restrictions on the choice of waveform dictionary $\mathcal{F}$. The invertibility of $\mathcal{F}$ is no longer an issue, and $\mathcal{F}$ may be *overcomplete*. This means that the number of columns, or dictionary vectors, may be larger than the dimension of $\mathbf{x}$. The motivation for allowing overcomplete dictionaries is simply to increase the choice of possible expansion vectors in the hope that this allows for better approximation capability while preserving the sparseness criterion.
2) Since we no longer assume that $\mathcal{F}$ is neither orthogonal nor invertible, we obviously have to find some other way of determining the expansion coefficients than what is done in traditional signal decompositions. We have employed the *fast orthogonal matching pursuit* (FOMP) algorithm of [2].

The main issue dealt with in this paper is as follows: Given a training signal denoted $\mathbf{x}$, or $x(n)$, *how do we find the waveform dictionary such that the best possible signal approximation results when keeping a small collection of coefficients?* In a practical setting, the training signal would contain different

signal realizations from the signal source for which we optimize [3]. Note that we do not impose any constraints requiring the $\mathbf{f}_i$s to be orthogonal; in fact, we also keep open the option of the set $\{\mathbf{f}_i\}$ being *linearly dependent*. Our only constraint is that $||\mathbf{f}_i|| = 1$ for all $i$.

The conceptually simplest choice of dictionary matrix is found when emulating the signal decomposition performed by a block transform. Denote by $\mathbf{F}$ the $N \times N$ matrix constituting the block transform. The columns of $\mathbf{F}$ are often referred to as the transform vectors. The associated waveform dictionary is given by

$$\mathcal{F} = \begin{bmatrix} \ddots & & & & \\ & \mathbf{F} & & & \mathbf{0} \\ & & \mathbf{F} & & \\ & \mathbf{0} & & \mathbf{F} & \\ & & & & \ddots \end{bmatrix}. \tag{4}$$

Several authors have investigated the use of overcomplete dictionaries in a sparse selection setup. In most cases, the focus has been on *block oriented* dictionaries, which are often referred to as *frames*. A frame is simply a transform where the number of transform vectors ($K$) is larger than the block length ($N$), i.e., an overcomplete transform. The frame structure is easily emulated in our setting by allowing $\mathbf{F}$ in (4) to have dimension $N \times K$, where $K > N$.

There is a relationship between vector quantization (VQ) and frame-based representation. In the extreme case where only *one* dictionary vector is chosen to represent a signal vector, we have a shape-gain VQ system. If the coefficient value is set to one and we allow the dictionary vectors arbitrary norm, we have a VQ system. In multistage VQ (MSVQ), the encoding is done in several stages, and the reconstructed vector is the sum of the vectors from the different stages; thus, MSVQ has obvious similarity with frame-based representation. However, the methods differ in both design and encoding procedures. Frame-based representation offers more flexibility in that a variable number of vectors can be used for representing different signal vectors. A thorough discussion on this topic can be found in [4].

Goyal and Vetterli have worked with frames or overcomplete expansions [5]–[7] using different frames. The frames they have used are chosen rather than optimized. As examples, they propose vectors on the $NM$-dimensional sphere that maximize the minimum Euclidean norm between the vectors, or corners, of the hypercube. In most cases, the research has focused on how to apply the frame in a data compression setup, whereas the question of frame *design* has received little attention. Some work in that area is done by Olshausen and Field [8] and Lewichy and Sejnowsky [9].

The present paper is a generalization of the block-based frame optimization algorithm of Engan *et al.* [3], [10], [11] to a much wider class of dictionary configurations, facilitating vectors overlapping *ad infinitum*, as in a filterbank or wavelet decomposition. Examples are shown in Fig. 1. We show how to optimize this class of decompositions with respect to sparseness.

The paper is organized as follows: Section II starts with a discussion on important choices of internal structure for the dictionary matrix $\mathcal{F}$ and relates them to well-known signal decomposition techniques. In the following subsections, we present an iterative algorithm for optimizing the waveform dictionary with respect to the sparseness criterion using a training signal. A detailed mathematical analysis shows that in each iteration, an improved dictionary can be found by solving a set of linear equations. Section III presents some results and examples showing the improved approximation capabilities of the designed waveform dictionaries as compared with that of more traditional decompositions. Finally, in Section IV, we conclude the paper and indicate some directions for further research.

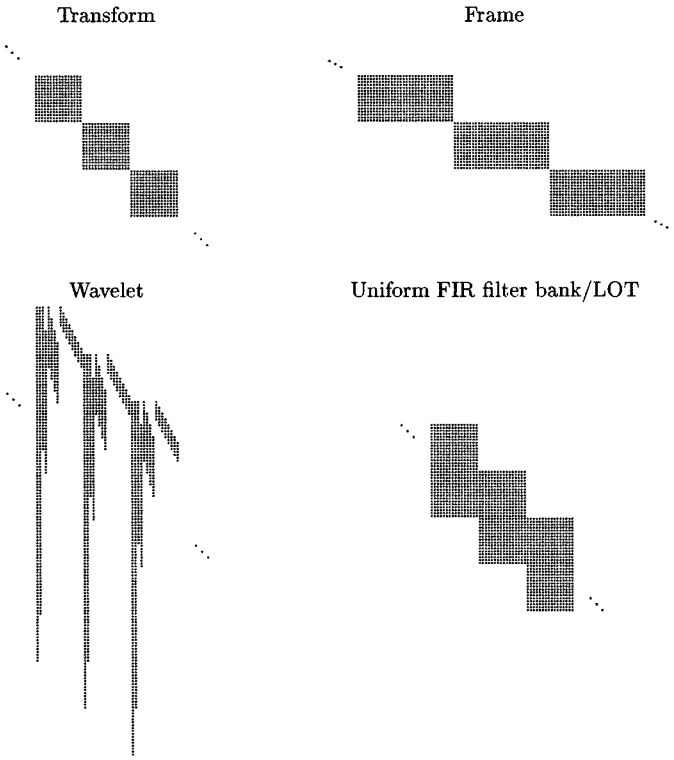


Fig. 1. Waveform dictionaries corresponding to traditional decompositions. Starting with the transform dictionary, the dot columns within each square represent the transform vectors. The frame dictionary is similar, but the number of frame vectors ($K$) is larger than the block length ($N$). In the filterbank/LOT case, the vectors are twice as long as in the transform case, thus rendering a 50% overlap between adjacent blocks. A wavelet uses dyadic frequency partitioning, resulting in different time shifts for expansion vectors corresponding to different frequency bands. This is seen in the figure where the (large) vectors corresponding to the low frequency bands have longer shifts than the vectors corresponding to higher frequency bands.

## II. Design Methodology

In the following, we focus our attention on signal expansions that can be seen as generalizations of transforms, frames, wavelets, wavelet packets, and filterbanks. This is motivated from the fact that these decompositions have been useful for compact signal representation and, thus, represent a starting point in our search for good choices of dictionaries. The idea is, therefore, to *postulate the internal structure of the dictionary matrix in a manner to mimic traditional signal decompositions in the hope that interesting signal expansions will result.*

Fig. 1 shows four different choices of $\mathcal{F}$ corresponding to traditional signal decompositions: transform, frame, wavelet, and

uniform FIR filter bank/lapped orthogonal transform (LOT). In each case, three identical blocks of the expansion vectors are shown. The dots signify nonzero entries of the dictionary matrix.

In Section III-B, these choices of $\mathcal{F}$ will be used as starting points for the optimization process. Note that the width of each sub-block within the dictionary ultimately decides whether the dictionary is undercomplete, complete, or overcomplete.

Looking again at the waveform dictionaries shown in Fig. 1, it is clear that the associated coefficient vector $\mathbf{w}$ will contain all expansion coefficients in the following manner: Using the transform case as an example, for each block of coefficients the ordering could be lowpass, bandpass, ..., bandpass, and highpass. For the wavelet case shown, the ordering would be one lowpass, one bandpass, two bandpass, four bandpass, and eight highpass coefficients for each block. In the derivations to follow, it is conceptually and notationally convenient to collect expansion coefficients used for expansion vector number $k$ (i.e., channel number $k$) into the vector $\mathbf{w}_k$. This corresponds to a simple permutation of the elements of $\mathbf{w}$ and the columns of $\mathcal{F}$, and we can write

$$\mathcal{F} = [\mathcal{F}_0 | \dots | \mathcal{F}_{K-1}] \tag{5}$$

and the signal expansion can be written as

$$\tilde{\mathbf{x}} = \mathcal{F}\mathbf{w} = \mathcal{F}_0\mathbf{w}_0 + \cdots + \mathcal{F}_k\mathbf{w}_k + \cdots + \mathcal{F}_{K-1}\mathbf{w}_{K-1} \tag{6}$$

where $\mathcal{F}_k$ contains the expansion vector corresponding to channel $k$ repeated *ad infinitum* using a time shift of $N_k$ samples:

$$\mathcal{F}_k = \begin{bmatrix} \ddots & & & \\ & f_k(0) & & \\ & \vdots & & \\ & f_k(N_k) & f_k(0) & \\ & \vdots & \vdots & \\ & f_k(L_k-1) & \vdots & \\ & & \vdots & \\ & & f_k(L_k-1) & \\ & & & \ddots \end{bmatrix}. \tag{7}$$

Using filterbank terminology, the signal expansion presented above is equivalent to a synthesis filterbank with $K$ channels, where channel number $k$ has filter length $L_k$ and upsampling factor $N_k$. The filter length of each channel $L_k$ is set to any number satisfying

$$L_k \geq N_k \tag{8}$$

which means that the filter length of each channel should be at least as long as the upsampling factor.

The methodology for designing a dictionary for sparse representation is as follows.

1) Choose a structure for $\mathcal{F}$ by selecting the values for $K$, the $N_k$s, and the $L_k$s.
2) Optimize $\mathcal{F}$ with respect to $f_k(j)$, $k = 0, \dots, K-1$, and $j = 0, \dots, L_k - 1$. The optimization criterion is minimum reconstruction error given a sparseness constraint on $\mathbf{w}$.

The optimization is performed with respect to a given signal $x(n)$, which in the derivations is assumed to be infinite. That is the topic of the following subsection.

### A. Dictionary Optimization

The iterative algorithm used to optimize a dictionary is inspired by the generalized Lloyd algorithm (GLA) used for designing vector quantization (VQ) codebooks [12]. Some of the details on how this algorithm motivates our own algorithm for transform vector design is reported in a somewhat different context in [11]. The key issue is that our algorithm, in a way similar to the GLA, is partitioned into two distinct parts.

1) $\mathcal{F}$ and $\mathbf{x}$ are known. Find a sparse coefficient vector $\mathbf{w}$.
2) $\mathbf{w}$ and $\mathbf{x}$ are known. Find the best possible $\mathcal{F}$.

Each iteration of the algorithm involves two parts: In the first part, we find approximations to the training signal using an initial dictionary or the dictionary resulting from the previous iteration. The second part tries to find a new dictionary in such a way that when used in conjunction with the *already computed* expansion coefficients, the residuals of the approximation in the first step is decreased. Denoting the training signal by $\mathbf{x}$, the structure of the algorithm can thus be stated as follows.

1) Begin with an initial dictionary $\mathcal{F}^{(0)}$, and fix a value for the *sparseness factor* defined as
$$S = \frac{\text{Number of nonzero coefficients in } \mathbf{w}}{\text{Number of samples in } \mathbf{x}}. \tag{9}$$
2) Approximate $\mathbf{x}$ as
$$\tilde{\mathbf{x}} = \mathcal{F}^{(0)}\mathbf{w} \tag{10}$$
where the sparseness constraint defined by $S$ is satisfied. Assign counter variable $i = 1$.
3) Given the current coefficient vector $\mathbf{w}$, find the optimal $\mathcal{F}$, which is denoted $\mathcal{F}^{(i)}$. Normalize dictionary vectors to unit length.
4) Using the new dictionary, find the new approximation. If $(\text{stop-criterion} = \text{FALSE}) \Rightarrow i = i + 1$, go to step 3. Otherwise, stop.

The main idea in the GLA is to reduce a complex optimization task into a main loop with two steps, where each step is solved in an optimal manner. By iteratively solving for the best coefficient vector, the best dictionary, and so forth, the approximation distortion is gradually decreased, and the algorithm guarantees convergence to a local optimum. Suggested stop-criteria can be maximum number of iterations, almost constant MSE, or complete convergence when that occurs.

In our algorithm, steps 3 and 4 constitute the main loop. As will be shown in Section II-B, step 3 guarantees reduced or unaltered distortion by finding the optimal dictionary for the coefficients chosen. However, in step 4, the best way of ensuring better vector approximations and, thus, a reduced overall distortion, is to perform a complete search of possible vector selections. As will be explained in Section II-C, this is computationally expensive, and we have to resort to vector selection algorithms. Due to suboptimal vector selection, it follows that the algorithm given above is not guaranteed to converge. The convergence properties are not yet fully understood. Due to the lack of guarantee for the new frame to be better than the previous, the algorithm

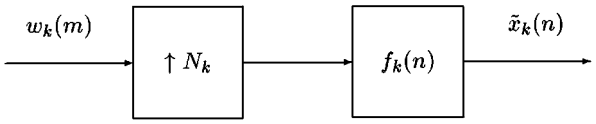


Fig. 2. Synthesis from channel number $k$.

should allow the MSE to grow for several iterations without terminating the training. This can be seen from a training example in Section III-A.

### B. Computing the Optimal Dictionary Update

We now explain how to perform the improvement in $\mathcal{F}$ constituting step 3 of the design algorithm. The objective is to find the $\mathcal{F}$ that minimizes the approximation residual while keeping the already-computed coefficients in the expansion resulting from step 2 (or 4) of the previous iteration of the algorithm. In the derivations to follow, the optimal $\mathcal{F}$ is found by simply differentiating the quadratic approximation error with respect to all elements of the dictionary vectors. For all cases considered, it is shown that this leads to a linear system of equations solvable for $\mathcal{F}$.

We start by solving the problem in its full generality using filter bank terminology. As explained in the introduction of Section II, (6) and (7) can be viewed as a multichannel synthesis system, where $\mathcal{F}_k\mathbf{w}_k$ denote the contribution from channel number $k$. This is illustrated in Fig. 2, where $w_k(m)$ are the elements of the vector $\mathbf{w}_k$ in (6), which is now interpreted as a time series.

The contribution from channel number $k$ to the synthesized signal can be written [13]

$$\tilde{x}_k(n) = \sum_{p=0}^{\lfloor (L_k-1)/N_k \rfloor} w_k\left(\left\lfloor \frac{n}{N_k} \right\rfloor - p\right) f_k(n \bmod N_k + pN_k) \tag{11}$$

where $\lfloor y \rfloor$ denotes the largest integer value less than, or equal to, $y$. The reconstructed signal is

$$\tilde{x}(n) = \sum_{k=0}^{K-1} \tilde{x}_k(n). \tag{12}$$

We define the object function $J$ as

$$J = \sum_{n=-\infty}^{\infty} (x(n) - \tilde{x}(n))^2 \tag{13}$$

and by setting its gradient to zero a straightforward derivation leads to

$$\begin{aligned}
&\sum_{n=-\infty}^{\infty} \sum_{k=0}^{K-1} \sum_{p=0}^{\lfloor (L_k-1)/N_k \rfloor} \sum_{t=0}^{K-1} \sum_{s=0}^{\lfloor (L_t-1)/N_t \rfloor} f_t(n \bmod N_t + sN_t) \\
&\quad \cdot w_k\left(\left\lfloor \frac{n}{N_k} \right\rfloor - p\right) w_t\left(\left\lfloor \frac{n}{N_t} \right\rfloor - s\right) \\
&\quad \cdot \frac{\partial f_k(n \bmod N_k + pN_k)}{\partial f_r(q)} \\
&\quad = \sum_{n=-\infty}^{\infty} \sum_{k=0}^{K-1} \sum_{p=0}^{\lfloor (L_k-1)/N_k \rfloor} x(n) w_k\left(\left\lfloor \frac{n}{N_k} \right\rfloor - p\right) \\
&\qquad \cdot \frac{\partial f_k(n \bmod N_k + pN_k)}{\partial f_r(q)}.
\end{aligned} \tag{14}$$

Here, we have differentiated with respect to $f_r(q)$, $r = 0, \ldots, K-1$, and $q = 0, \ldots, L_r - 1$.

The derivative term in (14) vanishes for all values of $n$, $k$, and $p$, except when

1) $k = r$;
2) $n \bmod N_r + pN_r = q$

in which case, it equals unity. Using the division algorithm due to Euclid, we may decompose $q$ as

$$q = \left\lfloor \frac{q}{N_r} \right\rfloor N_r + q \bmod N_r. \tag{15}$$

Comparing (15) with the expression in condition number 2 above, the uniqueness of the decomposition enforces $p = \lfloor q/N_r \rfloor$ and $n = q \bmod N_r + mN_r$, where $m$ is a new summation variable. Equation (14) now becomes

$$\begin{aligned}
&\sum_{t=0}^{K-1} \sum_{s=0}^{\lfloor \frac{L_t-1}{N_t} \rfloor} \sum_{m=-\infty}^{\infty} f_t((q \bmod N_r + mN_r) \bmod N_t + sN_t) \\
&\quad \cdot w_r\left(m - \left\lfloor \frac{q}{N_r} \right\rfloor\right) w_t\left(\left\lfloor \frac{q \bmod N_r + mN_r}{N_t} \right\rfloor - s\right) \\
&\quad = \sum_{m=-\infty}^{\infty} x(q \bmod N_r + mN_r) w_r\left(m - \left\lfloor \frac{q}{N_r} \right\rfloor\right).
\end{aligned} \tag{16}$$

*Left Side:* We now focus on the left-hand side of (16), where the goal is to remove the infinite summation over the $f_t(\cdot)$ factor with the objective of establishing a finite, linear set of equations in the $f_t(\cdot)$s. We change the summation variable as

$$m = Nu + v, \quad \text{where} -\infty < u < \infty, 0 \le v < N \tag{17}$$

where $N$ is the least common multiple of the upsampling factors:

$$N = \mathrm{LCM}(N_0, \ldots, N_{K-1}). \tag{18}$$

Using (17), we can write

$$\begin{aligned}
&\sum_{v=0}^{N-1} \sum_{u=-\infty}^{\infty} f_t((q \bmod N_r + uN_rN + vN_r) \bmod N_t + sN_t) \\
&\quad \cdot w_r\left(uN + v - \left\lfloor \frac{q}{N_r} \right\rfloor\right) \\
&\quad \cdot w_t\left(\left\lfloor \frac{q \bmod N_r + uN_rN + vN_r}{N_t} \right\rfloor - s\right) \\
&\quad = \sum_{v=0}^{N-1} f_t((q \bmod N_r + vN_r) \bmod N_t + sN_t) \\
&\qquad \cdot \sum_{u=-\infty}^{\infty} w_r\left(uN + v - \left\lfloor \frac{q}{N_r} \right\rfloor\right) \\
&\qquad \cdot w_t\left(\left\lfloor \frac{q \bmod N_r + vN_r}{N_t} \right\rfloor + uN_r\frac{N}{N_t} - s\right).
\end{aligned} \tag{19}$$

*Right Side:* Using the fact that

$$\sum_{m=-\infty}^{\infty} g(m+i)h(pm) = \sum_{m=-\infty}^{\infty} g(m)h(p(m-i)) \tag{20}$$

the expression on the right-hand side in (16) can be simplified as (21)

$$\sum_{m=-\infty}^{\infty} x\left(q \bmod N_r + \left\lfloor \frac{q}{N_r} \right\rfloor N_r + mN_r\right) w_r(m) = \sum_{m=-\infty}^{\infty} x(q+mN_r)w_r(m). \tag{21}$$

*Putting it All Together:*Collecting the results from (19) and (21), the optimal values $\{f_r(q)\}$ must satisfy

$$\sum_{t=0}^{K-1} \sum_{s=0}^{\lfloor \frac{L_t-1}{N_t} \rfloor} \sum_{v=0}^{N-1} f_t((q \bmod N_r + vN_r) \bmod N_t + sN_t) \cdot \sum_{u=-\infty}^{\infty} w_r\left(uN + v - \left\lfloor \frac{q}{N_r} \right\rfloor\right) \cdot w_t\left(\left\lfloor \frac{q \bmod N_r + vN_r}{N_t} \right\rfloor + uN_r\frac{N}{N_t} - s\right) = \sum_{m=-\infty}^{\infty} x(q+mN_r)w_r(m) \tag{22}$$

for all values $r = 0, \ldots, K-1$, and $q = 0, \ldots, L_r - 1$.

A matrix formulation equivalent to (22) is obtained by setting

$$a(r,q,t,s,v) = \sum_{u=-\infty}^{\infty} w_r\left(uN + v - \left\lfloor \frac{q}{N_r} \right\rfloor\right) \cdot w_t\left(\left\lfloor \frac{q \bmod N_r + vN_r}{N_t} \right\rfloor + uN_r\frac{N}{N_t} - s\right) \tag{23}$$

$$b_{rq} = \sum_{m=-\infty}^{\infty} x(q+mN_r)w_r(m) \tag{24}$$

$$g(r,q,t,s,v) = (q \bmod N_r + vN_r) \bmod N_t + sN_t \tag{25}$$

$$\mathcal{A}_{ti}^{rq} = \sum_{\{(s,v)|g(r,q,t,s,v)=i\}} a(r,q,t,s,v) \tag{26}$$

$$\mathbf{f} = (f_0(0), f_0(1), \ldots \ldots, f_{K-1}(L_{K-1}-2), f_{K-1}(L_{K-1}-1))^{\mathrm{T}} \tag{27}$$

giving

$$\begin{bmatrix} \mathcal{A}_{00}^{00} & \mathcal{A}_{01}^{00} & \cdots & \mathcal{A}_{K-1,L_{K-1}-1}^{00} \\ \mathcal{A}_{00}^{01} & \mathcal{A}_{01}^{01} & \cdots & \mathcal{A}_{K-1,L_{K-1}-1}^{01} \\ & \vdots & & \\ \mathcal{A}_{00}^{K-1,L_{K-1}-1} & \mathcal{A}_{01}^{K-1,L_{K-1}-1} & \cdots & \mathcal{A}_{K-1,L_{K-1}-1}^{K-1,L_{K-1}-1} \end{bmatrix} \cdot \mathbf{f} = \begin{bmatrix} b_{00} \\ b_{01} \\ \vdots \\ b_{K-1,L_{K-1}-1} \end{bmatrix}. \tag{28}$$

From this linear set of $\sum_{r=0}^{K-1} L_r$ equations, we can compute the vector $\mathbf{f}$ constituting the best possible dictionary $\mathcal{F}$, in the mean square error sense, when the coefficients for each vector approximation are fixed. Since the object function (13) is bounded downwards, we know that the problem has a minimum solution. Since (22) only has one solution when the linear system has full rank, this must be the unique, global solution.

Notice again the generality of (22) and (28). A vast set of different configurations may be optimized using the formulation above, including cases where the dictionary is under or overcomplete.

In the following subsections, we study three very important special cases and their generalizations:

1) Block transforms/frames;
2) uniform filter banks/LOTs
3) wavelet-like expansions.

*1) Special Case Number 1: Block Transforms/Frames:* If we set $N_0 = N_1 = \cdots = N_{K-1} = N$ and $L_0 = L_1 = \cdots = L_{K-1} = N$, we obtain a block-oriented signal expansion. If $K = N$, we have a transform, and if $K > N$, we have a block-oriented frame.

Equation (22) can now be written as

$$\sum_{t=0}^{K-1} f_t(q) \sum_{m=-\infty}^{\infty} w_r(m)w_t(m) = \sum_{m=-\infty}^{\infty} x(q+mN)w_r(m) \tag{29}$$

where $r = 0, \ldots, K-1$ and $q = 0, \ldots, N-1$.

Defining the $N \times K$ matrix $\mathbf{F}$ as

$$\mathbf{F} = \begin{bmatrix} f_0(0) & f_1(0) & \cdots & f_{K-1}(0) \\ f_0(1) & f_1(1) & \cdots & f_{K-1}(1) \\ & & \vdots & \\ f_0(N-1) & f_1(N-1) & \cdots & f_{K-1}(N-1) \end{bmatrix} \tag{30}$$

and the matrices $\mathbf{A}$ and $\mathbf{B}$ by their elements as

$$A_{rt} = \sum_{m=-\infty}^{\infty} w_r(m)w_t(m) \tag{31}$$

$$B_{qr} = \sum_{m=-\infty}^{\infty} x(q+mN)w_r(m) \tag{32}$$

(29) can be written as

$$\mathbf{FA} = \mathbf{B} \tag{33}$$

with the solution

$$\mathbf{F} = \mathbf{BA}^{-1}. \tag{34}$$

This is equivalent to the result derived in [3] and [11].

*2) Special Case Number 2: Uniform Filter Banks/LOTs:* If we set $N_0 = N_1 = \cdots = N_{K-1} = N$ and $L_0 = L_1 = \cdots = L_{K-1} = PN$, where $P$ is a positive integer, we obtain a signal expansion similar to a uniform, $K$-channel, FIR filterbank where the channels have the same filter length. This is sometimes referred to as a LOT [14], [15].

Using the division algorithm due to Euclid, we can decompose $q$ as

$$q = q_1 + q_2N \tag{35}$$

where $q_1 = q \bmod N$, and $q_2 = \lfloor q/N \rfloor$. Equation (22) then becomes

$$\sum_{t=0}^{K-1}\sum_{s=0}^{P-1} f_t(q_1+sN) \underbrace{\sum_{m=-\infty}^{\infty} w_r(m-q_2)w_t(m-s)}_{a(t,s,q_2,r)} = \underbrace{\sum_{m=-\infty}^{\infty} x(q_1+q_2N+mN)w_r(m)}_{b(q_1,q_2,r)} \tag{36}$$

where $r = 0,\ldots,K-1, q_1 = 0,\ldots,N-1, q_2 = 0,\ldots,P-1$.

Defining the $N \times PK$ matrix $\mathbf{F}^*$ as (37), shown at the bottom of the page, and the matrices $\mathbf{A}$ and $\mathbf{B}$ by their elements as

$$\begin{aligned} A_{kl} &= a\left(\left\lfloor \frac{k}{P}\right\rfloor, k \bmod P, l \bmod P, \left\lfloor \frac{l}{P}\right\rfloor\right) \\ k,l &= 0,\ldots,PK-1 \\ B_{kl} &= b\left(k, l \bmod P, \left\lfloor \frac{l}{P}\right\rfloor\right) \\ k &= 0,\ldots,N-1, \quad l = 0,\ldots,PK-1 \end{aligned} \tag{38}$$

Equation (36) can be written as

$$\mathbf{F}^*\mathbf{A} = \mathbf{B} \tag{39}$$

with the solution

$$\mathbf{F}^* = \mathbf{B}\mathbf{A}^{-1}. \tag{40}$$

Note that the $\mathbf{F}^*$ matrix is simply introduced in order to provide a compact formulation for the solution of the optimal dictionary in the uniform FIR filterbank case and does not relate directly to the internal structure of the dictionary matrix itself. We would also like to emphasize the possibility of optimizing configurations where $K \neq N$.

*3) Special Case Number 3: Wavelets:* A discrete-time wavelet decomposition is constructed by repeatedly using two filters (lowpass and highpass) on the output of the lowpass filter in each stage. In this manner, a dyadic frequency partitioning results [13].

The general framework outlined in Section II-B can easily be adapted to mimic a general, $M$-stage, wavelet-type expansion by setting $K = M+1$ and the upsampling parameters as follows:

$$\begin{aligned} N_0 &= 2^M \text{ ``Lowpass channel"} \\ N_1 &= 2^M \text{ ``1. bandpass channel"} \\ N_2 &= 2^{M-1} \text{ ``2. bandpass channel"} \\ N_3 &= 2^{M-2} \text{ ``3. bandpass channel"} \\ &\vdots \\ N_M &= 2 \text{ ``Highpass channel".} \end{aligned} \tag{41}$$

The optimal filters are found by inserting the chosen upsampling parameters and filter lengths into (28) and setting $N = 2^M$. A varied set of under or overcomplete wavelet-like signal expansions can be generated in the same manner. Note that the resulting expansions are not wavelets but expansions mimicking the *support structure* of wavelets.

### C. Finding a Signal Approximation

As explained in the introduction, by giving up the invertibility of the dictionary matrix $\mathcal{F}$, we have to find some other way of computing the expansion coefficients of (3). The goal is to use few coefficients while constructing a good approximation to the training signal $x(n)$ or $\mathbf{x}$.

Ideally, when approximating the signal $x(n)$, *the available coefficients should be allocated over the whole signal.* To illustrate this idea, let us focus on the simple case where the dictionary is a block-oriented signal expansion, where the $N \times K$ matrix $\mathbf{F}$ is defined as in (30) in Section II-B1. Furthermore, we assume the training signal to be finite, with length $L_x = MN$, where $M$ is a positive integer.

The signal expansion can be written using $\mathbf{F}$ repeated $M$ times, as illustrated in

$$\begin{bmatrix} \mathbf{F} & \mathbf{0} & \cdots & & \mathbf{0} \\ \mathbf{0} & \mathbf{F} & & & \\ \vdots & & \ddots & & \\ & & & \mathbf{F} & \mathbf{0} \\ \mathbf{0} & & & \mathbf{0} & \mathbf{F} \end{bmatrix} \begin{bmatrix} \mathbf{w}^{(0)} \\ \mathbf{w}^{(1)} \\ \vdots \\ \mathbf{w}^{(M-1)} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{x}}^{(0)} \\ \tilde{\mathbf{x}}^{(1)} \\ \vdots \\ \tilde{\mathbf{x}}^{(M-1)} \end{bmatrix} \tag{42}$$

where $\mathbf{x}^{(i)} \in R^N$ and $\mathbf{w}^{(i)} \in R^K$ denote the $i$th block of the signal and coefficient vector, respectively.

If the available coefficients are allocated freely among the $M$ $\mathbf{F}$-blocks, signal blocks with high energy content can be approximated using more coefficients than blocks having less energy. This corresponds to the practical situation occurring in straightforward transform coding when the vectors to be retained in the expansion are found by keeping those coefficients above a given threshold. Clearly, if $M$ is small, this scheme would offer little flexibility for approximating signal regions having varying characteristics.

Given a sparseness factor $S$ defined as in (9) and with practical considerations in mind, the global procedure of selecting $L_xS$ out of $L_xK/N$ vectors is prohibitive when $L_x$ is large, as will be the case in our experiments. It follows that the procedure of selecting vectors (and correspondingly, expansion coefficients) must be performed in a segment-by-segment manner.

We partition the training signal $x(n)$ into equally sized segments, and, depending on whether the dictionary is block-based or not, the procedure is performed as follows:

*Block-Based Dictionary:* Using a segment size equal to $MN$, each segment of $x(n)$ will be *independently* approximated, as

$$\mathbf{F}^* = \begin{bmatrix} f_0(0) & f_0(N) & \ldots & f_0((P-1)N) & \cdots & f_{K-1}(0) & \ldots & f_{K-1}((P-1)N) \\ f_0(1) & f_0(N+1) & \ldots & f_0((P-1)N+1) & \cdots & f_{K-1}(1) & \ldots & f_{K-1}((P-1)N+1) \\ & & \vdots & & & \vdots & & \\ f_0(N-1) & f_0(2N-1) & \ldots & f_0(PN-1) & \cdots & f_{K-1}(N-1) & \ldots & f_{K-1}(PN-1) \end{bmatrix} \tag{37}$$

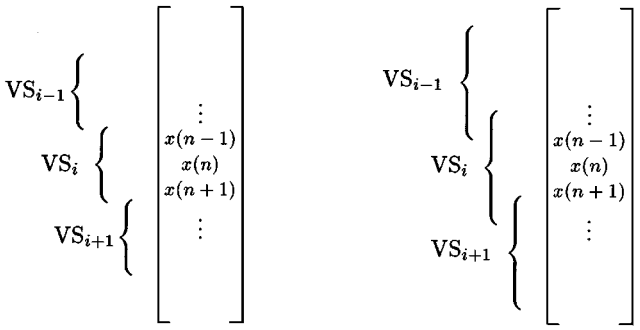


Fig. 3. Vector selection (VS) is performed segment-by-segment according to dictionary type: Block-based (left) and overlap (right).

shown in (42), using $MNS$ vectors. The procedure is shown on the left side of Fig. 3.

*Dictionary with Overlapping Vectors:* If the dictionary has overlapping vectors (for example, the uniform filterbank case in Section II-B2 or the wavelet case in Section II-B3), a perfect separation of the segment-based expansions is not possible. Due to overlapping vectors, adjacent expansions will overlap.

A practical way of overcoming the overlap problem is as follows: The training signal $x(n)$ is approximated segment-by-segment, in a top-down manner, as shown on the right side of Fig. 3. When the expansion for the current segment has been found, the associated reconstruction is subtracted from $x(n)$ before starting the procedure on the next segment.

*1) Vector Selection Algorithms:* For each segment of $x(n)$, the task is to find the best possible approximation using a predetermined number of dictionary vectors. If we wanted to find the *optimal* approximation to a given segment by selecting $a$ out of $b$ dictionary vectors, it would be necessary to investigate

$$\binom{b}{a} \tag{43}$$

different choices of vectors. Thus, finding the optimal vectors to use in an approximation is an NP-hard problem and requires extensive calculation [16]. It follows that a suboptimal algorithm is preferable in order to limit the computational complexity. There exist several different vector selection schemes for solving this problem [1], [17]–[19]. In this paper, the *fast orthogonal matching pursuit* (FOMP) algorithm of [2] is used, but we stress that other vector selection schemes also can work within the dictionary design framework.

## III. Design Examples and Discussion

The framework presented can be used for optimizing virtually any choice of expansion configuration. In this section, we focus on the following established expansions:

- Block transforms.
- frames;
- uniform FIR filterbanks;
- wavelets.

For each case, we will pick a representative expansion and demonstrate that significant improvements are possible using the optimization technique presented earlier. The improvement is measured in terms of increased approximation capability for a given choice of $S$, which is the sparseness factor. To emulate a low bit rate situation, the value used for $S$ always belong to the set $\{1/4, 1/6, 1/8, 1/10\}$, which means that on average only one expansion coefficient out of four, six, eight, or ten signal samples is kept.

In all experiments, the signal $x(n)$ to be represented is a realization of a Gaussian AR(1) process with correlation factor $\rho = 0.95$ and with unit variance. The training signal length should be large in order to ensure proper generalization. This means that the optimized signal expansion should give similar approximation results when used on a new realization of the AR(1) process. Here, a signal length of 204 800 samples proved to be sufficient.

### A. Improving the Karhunen–Loéve Transform (KLT)

Although widely considered optimal for compact representation of the chosen Gaussian AR(1) process, it should be emphasized that the KLT only guarantees minimum distortion when used in conjunction with *linear* approximations, i.e., when always retaining the *first* coefficients of a transform block. Optimality is not ensured when the scheme for picking coefficients is based on importance rather than position [20].

Using the $N = 16$ point KLT as a starting point for the optimization scheme, the resulting waveform dictionary $\mathcal{F}$ is as shown in the upper left corner of Fig. 1. The goal is now to modify the basis vectors of the KLT in such a manner that the approximating power is increased. In this experiment, the sparseness factor is $S = 1/4$, and in each optimization iteration, the coefficients are found using vector selection on a segment of size $16 \times N = 256$ samples, meaning that 64 out of 256 transform vectors will be selected. This corresponds to a bit allocation situation where a pool of bits are distributed among the coefficients belonging to 16 adjacent transform blocks.

Fig. 4 depicts the obtained training curve showing the approximation error as a function of the number of training iterations. The distortion in iteration 1 is that obtained using the KLT. As explained in Section II-A, each iteration involves two parts: First, find an approximation using vector selection and then compute the best possible waveform dictionary for that set of coefficients. For about 30 iterations, the improvement is monotonous, and then there is a slight increase. The increase in the approximation distortion is a result of the suboptimal vector selection.

The dotted, horizontal line in the plot shows the obtained approximation distortion when retaining the *optimal* set of coefficients/vectors of the KLT. The orthogonal properties of the KLT facilitates easy computation of the best coefficients using the analysis transform. By simply selecting the largest coefficients, the optimal selection is done. In iteration number 1, where the KLT is used as initial transform, we observe that the distortion increase due to suboptimal vector selection is marginal. Furthermore, using the optimized expansion in conjunction with suboptimal vector selection the distortion is reduced by about 15%.

When viewing a traditional transform coder as a subband coder, it is well known that the subband channel responses cover the whole frequency range in a uniform manner [21]. In Fig. 5, we have plotted the amplitude responses for the subband interpretation of a transform coder when using the transform vectors designed above. We observe that the channel responses for

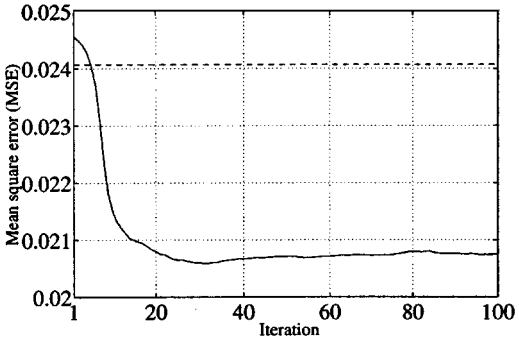


Fig. 4. Training curve using the 16-point KLT as the initial transform and keeping one out of four expansion coefficients. The input signal is a realization of an AR(1), $\rho = 0.95$ Gaussian process. The mean-square approximation error (MSE) is plotted as a function of the iteration number.

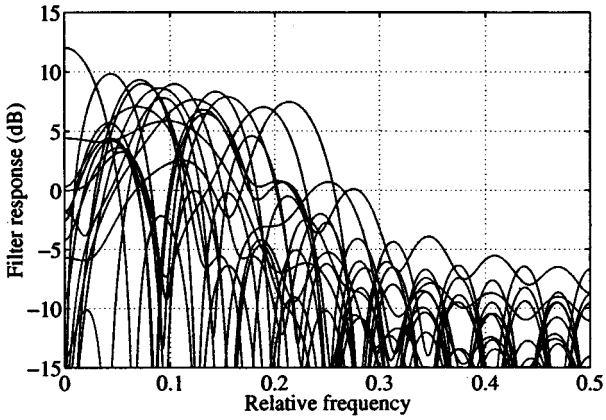


Fig. 5. Frequency responses of the $N = 16$ channels corresponding to a filterbank interpretation of the optimized transform matrix.

the low-frequency range is very similar to what we would obtain with a DCT or a KLT but that there are no channels in the very-high-frequency range. Given the high degree of sparseness we are aiming at here, this is logical. Rather than computing both high- and low-frequency transform coefficients and then throwing away the high-frequency coefficients, as is often done in classical low bit rate transform coders, we end up with a flexible set of transform vectors, corresponding to overlapping frequency ranges and designed for the best possible approximation of the signal at hand.

We would like to stress that the test of this section is rather harsh. For example, there are no nonstationarities in the examples above to be exploited, but still, the KLT is outperformed. In fact, a similar, or even better performance increase than the one reported above can be realized on real-life nonstationary signals, such as images, speech, and electrocardiogram (ECG) signals. For the case of ECG signals, this was demonstrated in [22].

In a data compression setup, not only the (quantized) coefficients, but also their *position*, would have to be transmitted. Due to the lowpass characteristic of all the basis vectors of the optimized transform, we may anticipate a more uniform distribution of positions than in the KLT case. If entropy coding is used for bit-efficient representation of the position information, the optimized transform will give a higher bit rate for the *position* information. This is not the case if a fixed bit rate coding scheme is used. An investigation into these matters is a topic of future research.

### B. Optimizing Other Configurations

The previous section investigated the optimization of a block transform with the KLT as starting point. The obtained transform was optimized for a sparseness factor of $S = 1/4$. Using a different sparseness factor, a different transform would result.

In this section, we focus on approximation capability as a function of the sparseness factor. In the experiments, we use four different signal expansions as starting points:

*Transform:* The $N = 16$ point KLT, as in Section III-A.

*Frame:* We construct a $16 \times 32$ frame by merging the basis vectors from an $N = 16$ point discrete-cosine-transform (DCT) with an $N = 16$ point Haar transform. This frame was used in [23]–[25] in a data representation setup. The first basis vector of these two transforms are identical, and we substitute one of them with a unit-length random vector.

*Wavelet:* We constructed a four-stage dyadic decomposition using the biorthogonal 7/9 wavelet due to Cohen *et al.* [26]. Synthesis of reconstructed signals was done using the nine-tap lowpass filter and the seven-tap highpass filter.

*Uniform FIR Filterbank/LOT:* A 32-tap LOT with $N = 16$ channels is constructed. This filterbank is optimized for an AR(1) $\rho = 0.95$ process using an eigenvalue formulation [14].

The exact layout of the associated waveform dictionary $\mathcal{F}$ for each of these configurations is illustrated in Fig. 1.

Using the same AR(1) training signal as in the previous section, each of the four configurations described above was optimized using the original transform/filterbank coefficients as starting point for the iterations. Each configuration was optimized for $S = 1/4$, $1/6, 1/8$, and $1/10$. As the training curve was not always monotonic, the resulting decomposition was picked as the best one resulting from 100 iterations, except for the frame optimization, where 200 iterations were used due to slower convergence. The approximation capabilities of the obtained decompositions are plotted in Fig. 6 and compared with that of the original decompositions. All plots show the resulting approximation error when using a *new* realization of the AR(1) process. In the cases where the initial decompositions are unitary (KLT and LOT), optimal vector selection is used for the reference curves by retaining the largest coefficients generated using the corresponding analysis decomposition.

For all the chosen configurations, the optimized signal expansions outperform their original counterparts to varying degrees. We observe that the improvement due to optimization increases with the degree of sparseness. The main reason for this is as explained in Section III-A. For very sparse representations, the available expansions vectors should form a flexible set of lowpass vectors. The basis vectors of the KLT corresponding to the higher band channels are hardly used when $S$ is small. In addition, the use of a greedy vector selection algorithm also contributes to this trend because the vector selection will be closer to optimality when few vectors are selected.

For the three critically sampled configurations, the reduction in approximation error is roughly 30% when one out of ten vec-

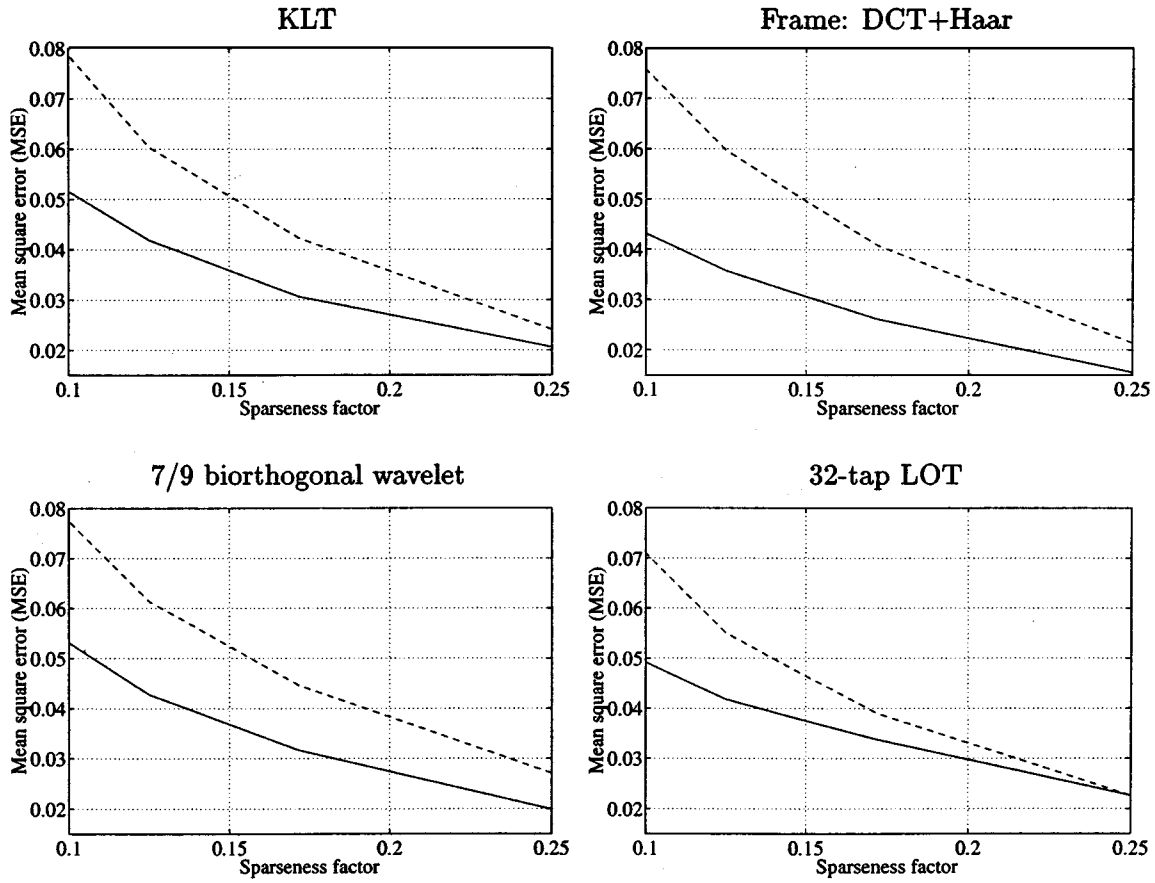


Fig. 6. Comparing the approximation capabilities of established signal expansions (dotted line) with identical configuration where the expansion vectors have been optimized (solid line).

tors are selected, whereas in the frame case, the reduction is about 40%. In addition, we observe that for each sparseness factor, the optimized frame attains the lowest approximation error of the four configurations tested.

## IV. Conclusions

In this paper, we have demonstrated that the approximating power of established signal decompositions like the KLT, the original LOT, and the 7/9 biorthogonal wavelet can be improved upon if the objective is to find an expansion of the signal with a given number of terms. This is achieved by giving up the analysis–synthesis paradigm in favor of a simple synthesis approach, where signal expansions are constructed using vector selection algorithms rather than using a corresponding analysis decomposition.

In addition to optimizing signal expansions where the positioning of the expansions vectors are set in order to mimic transforms, filterbanks, or wavelets, the presented algorithm can be used to optimize the approximation capabilities of virtually any kind of signal expansion, including over or undercomplete expansions. We feel that overcomplete expansions show promise for compact representation. This was demonstrated in the experiments where the optimized (overcomplete) frame outperformed the optimized transform in terms of approximation error. However, in a complete data compression setup, the improved approximation capability will have to be paid for by increased side information giving the positions of the coefficients used. This issue is a topic for further research.

The optimization algorithm is inspired by the GLA used for the design of vector quantizers and involves a training set embedding the properties of the class of signals under consideration. Because the vector selection algorithm is not optimal, the algorithm does not guarantee monotonous decrease of the MSE. In fact, the convergence properties of our design algorithm are not fully understood and is the subject of further investigations.

## References

[1] B. D. Rao, "Signal processing with the sparseness constraint," in *Proc. ICASSP*, Seattle, WA, May 1998, pp. 1861–1864.

[2] M. Gharavi-Alkhansari and T. S. Huang, "A fast orthogonal matching pursuit algorithm," in *Int. Conf. Acoust. Speech Signal Process.*, Seattle, WA, May 1998, pp. 1389–1392.

[3] K. Engan, S. O. Aase, and J. H. Husøy, "Multi-frame compression: Theory and design," *Signal Process.*, vol. 80, pp. 2121–2140, Oct. 2000.

[4] K. Engan, "Frame based signal representation and compression," Ph.D. dissertation, Norges teknisk-naturvitenskapelige universitet (NTNU)/Høgskolen i Stavanger, Stavanger, Norway, 2000.

[5] V. K. Goyal, "Quantized overcomplete expansions: Analysis, synthesis and algorithms, tech. rep," Electron. Res. Lab., memo. UCB/ERL M95/97, 1995.

[6] V. K. Goyal, M. Vetterli, and N. T. Thao, "Quantization of overcomplete expansions," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 1995, pp. 13–22.

[7] ——, "Quantized overcomplete expansions in RN: Analysis, synthesis, and algorithms," *IEEE Trans. Inform. Theory*, vol. 44, pp. 16–31, Jan. 1998.

[8] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed in V1," *Vis. Res.*, vol. 37, pp. 3311–3325, 1997.
[9] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," *Neural Comput.*, vol. 12, pp. 337–365, Feb. 2000.
[10] K. Engan, S. O. Aase, and J. H. Husøy, "Designing frames for matching pursuit algorithms," in *Proc. ICASSP* , Seattle, WA, May 1998, pp. 1817–1820.
[11] ——, "Method of optimal directions for frame design," in *Proc. ICASSP* , Phoenix, AZ, Mar. 1999, pp. 2443–2446.
[12] A. Gersho, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1992.
[13] T. A. Ramstad, S. O. Aase, and J. H. Husøy, *Subband Compression of Images -Principles and Examples*. Amsterdam, The Netherlands: Elsevier, 1995.
[14] H. S. Malvar and D. H. Staelin, "The LOT: Transform coding of images without blocking effects," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 553–559, Apr. 1989.
[15] H. Malvar, *Signal Processing with Lapped Transforms*. Norwell, MA: Artech House, 1992.
[16] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, pp. 227–234, Apr. 1995.
[17] S. S. Chen, "Basis Pursuit," Ph.D. dissertation, Stanford University, 1995.
[18] G. Davis, "Adaptive nonlinear approximations," Ph.D. dissertation, New York Univ., New York, 1994.
[19] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm," *IEEE Trans. Signal Processing*, vol. 45, pp. 600–616, Mar. 1997.
[20] S. Mallat and F. Falzon, "Understanding image transform codes," in *Proc. SPIE Aerosp. Conf.*, Orlando, FL, Apr. 1997.
[21] A. N. Akansu and R. A. Haddad, *Multiresolution Signal Decomposition*. San Diego, CA: Academic, 1992.
[22] J. H. Husøy, S. O. Aase, K. Skretting, and K. Engan, "Design of general block oriented expansions for efficient signal representation," in *Proc. ISCAS*, Orlando, FL, June 1999, pp. III–9–III–12.
[23] W. Mikhael and A. Ramaswamy, "Application of multitransforms for lossy image representation," *IEEE Trans. Circuits Syst. II*, vol. 41, pp. 431–434, June 1994.
[24] A. Berg and W. Mikhael, "Signal representation using adaptive parallel mixed transform techniques," in *Proc. 38th IEEE Midwest Symp. Circuits Syst.*, Aug. 1995.
[25] W. Mikhael and A. Berg, "Image representation using nonorthogonal basis images with adaptive weight optimization," *IEEE Signal Processing Lett.*, vol. 3, pp. 165–167, June 1996.
[26] A. Cohen, I. Daubechies, and J. C. Feauveau, "Biorthogonal bases for compactly supported wavelets," *Commun. Pure Appl. Math*, 1992.

**Sven Ole Aase** was born in Stavanger, Norway, in 1965. He received the M.Sc. and Ph.D. degrees in 1989 and 1993, respectively, both from The Norwegian Institute of Technology, Trondheim, Norway.

He is currently a professor with the Department of Electrical and Computer Engineering, Stavanger University College, Stavanger, Norway. His research interests include signal compression and representation, filterbank optimization, adaptive filters, and medical applications of digital signal processing.

**John Håkon Husøy** was born in Toronto, ON, Canada, in 1956. He received the M.Sc. and Ph.D. in electrical engineering in 1981 and 1991, respectively, from the Norwegian Institute of Technology, University of Trondheim, Trondheim, Norway.

He has been involved in hardware and software development in various positions in several companies. He is currently a professor with the Department of Electrical and Computer Engineering, Stavanger University College, Stavanger, Norway. His research interests include image compression, digital filtering, bioelectrical signal processing, adaptive algorithms, and image analysis.

**Karl Skretting** was born in Naerbø, Norway, in 1962. He received the B.Sc. degree in 1985 from Stavanger University College (SUC), Stavanger, Norway. He studied signal processing at Stavanger University College in 1996 and received the M.Sc. degree from the Department of Electrical and Computer Engineering in 1998. Currently, he is a research fellow and is pursuing the Ph.D. degree at SUC with the Signal Processing Group.

His research interests include signal representation and data compression.

**Kjersti Engan** was born in Bergen, Norway, in 1971. She received the B.Sc. degree in electrical engineering from Bergen University College in 1994 and the M.Sc. and Ph.D. degrees, both in electrical engineering, from Stavanger University College (SUC), Stavanger, Norway, in 1996 and 2000, respectively.

She is currently an Associate Professor with the Department of Electrical and Computer Engineering, at SUC. Her research interests include signal and image representation and compression.

Dr. Engan is a member the Norwegian Signal Processing Society.