ECG COMPRESSION USING SIGNAL DEPENDENT FRAMES AND MATCHING PURSUIT

Karl Skretting, Kjersti Engan and John Håkon Husøy

University of Stavanger
Department of Electrical and Computer Engineering
N-4036 Stavanger, Norway, E-mail: karl.skretting@uis.no

ABSTRACT

This paper describes a general design method for signal dependent frames, or over-complete dictionaries. The design method is developed for block-oriented frames, overlapping frames, and constrained frames, which can be regarded as extensions of respectively block-oriented transforms, filter banks, and general filter banks like wavelets. The frames are optimized for sparse representation of a signal class. Using the designed frames and a Matching Pursuit algorithm it is shown that superior, compared to other methods, sparse representations are achieved for an Electrocardiogram signal. It is also demonstrated that compression based on these sparse representations gives excellent performance at low bit rates.

1. INTRODUCTION

Compression of Electrocardiogram (ECG) signals is an important problem that has received considerable attention [1, 2, 3]. ECG signals are (recorded) electrical pulses the body makes during heart beats. In this paper we use signals from the MIT database [4], sampled at 360 Hz with 12 bits per sample.

Traditional compression schemes have the following steps: 1) The signal is decomposed into expansion coefficients by a transform, which may be a block-transform, a filter bank or a wavelet. The number of coefficients is the same as the number of signal samples, i.e. the decomposition method is *complete*. 2) The expansion coefficients are quantized. A common method is to use a uniform quantizer with thresholding, which makes the zero bin larger than the other bins. This has been proven to give good performance when used in combination with an End-Of-Block (EOB) coding scheme. At low bit rates many of the coefficients fall into the zero bin and are quantized to zero. The number of non-zero coefficients may be considerably lower than the original number of signal samples, giving a *sparse* representation. To quantify the degree of sparseness the *sparseness factor*, S, is defined as the proportion of non-zero coefficients in the signal expansion to the number of samples in the signal. 3) Entropy coding packs the quantized expansion coefficients in an efficient way, using few bits. This may by be done by quite simple Huffman coding or more complex bit-plane schemes, like in JPEG 2000. 4) Inverse entropy coding and inverse quantizing restore the quantized coefficients, i.e. they are approximations to their original values. 5) The reconstructed signal is built from the quantized expansion coefficients by the inverse of the transform used in step 1. This corresponds to forming the reconstructed signal as a linear combination of *synthesis vectors*. In a filter bank context this step is also called the *synthesis* step. This compression scheme is also used as a reference in the experiments in this paper.

Denoting the signal as a vector $\mathbf{x}$, the expansion coefficients are found as $\mathbf{y} = \mathcal{T}^{-1}\mathbf{x}$ where $\mathcal{T}^{-1}$ is a large (or even infinite) invertible, possibly orthogonal, matrix representing the transform. The quantized coefficients are denoted $\tilde{\mathbf{y}}$, and the reconstructed signal is formed by the *synthesis equation*:

$$\tilde{\mathbf{x}} = \mathcal{T}\tilde{\mathbf{y}}. \tag{1}$$

The columns of $\mathcal{T}$ are the synthesis vectors. Using block-oriented transforms the synthesis vectors within each block do not overlap with the synthesis vectors of other blocks and $\mathcal{T}$ is a block-diagonal matrix. For filter banks and wavelets the synthesis vectors will overlap with those in neighboring blocks, so filter banks and wavelets are *overlapping* transforms.

In [5], it is observed that at low bit rates the main factor determining the performance of an orthogonal transform coder is the approximation error caused by the sparseness. The more sparse the approximated representation of the signal is, the lower bit rate is achieved. Inspired by this observation, it would be an idea to seek for the transform that gives the best approximation for a given sparseness factor. Removing the constraint that the matrix $\mathcal{T}$ should be a square matrix and making its width larger than its height, i.e. K columns for every N rows and $K > N$, the matrix will no longer represent a transform but a *frame*, assuming that its columns span the space. The frame is denoted $\mathcal{F}$, it is *over-complete* and the degree of over-completeness is K/N. Using a frame instead of a transform will increase the number of synthesis vectors available and finding a better approximation should be possible using the same sparseness factor. In this paper we present a design method for signal dependent frames, which, for a given sparseness factor, searches for a frame where the representation error of a training signal is minimized. We will here show that these frames not only give better sparse representations, but also give better compression results at low bit rates.

Since we assume that $\mathcal{F}$ is neither orthogonal nor invertible we can not find the coefficients in the same way as in traditional transform coders. We will not look for some quantized coefficients, $\tilde{\mathbf{y}}$, but search directly for an appropriate sparse vector of weights, $\mathbf{w}$, that can be used in the synthesis equation,

$$\tilde{\mathbf{x}} = \mathcal{F}\mathbf{w}. \tag{2}$$

The problem of finding the sparse weight vector, for a given sparseness factor, such that the 2-norm of the residual is minimized, is an NP-hard problem [6]. Many practical solutions employ greedy *vector selection* algorithms such as Matching Pursuit (MP), Orthogonal Matching Pursuit (OMP) and Order Recursive Matching Pursuit (ORMP) [7, 8]. ORMP is used in this work.

2. FRAME DESIGN

The design algorithms of this section constitute a synthesis of techniques presented at various levels of development in [9, 10]. The algorithms presented aim at training a frame to give a good, sparse representation of a class of signals. The optimal frame will depend on the target sparseness factor and the class of signals we want to represent. Thus, we need representative data as training data. The training signal, $\mathbf{x}$, is divided into L blocks, $\mathbf{x}_l$, each of length N. The training signal blocks, and also the weight blocks and the frame itself, will be organized in many different ways in the cases presented below, whatever is appropriate for the situation.

2.1. Unconstrained Block Based Frame Design

For a one dimensional signal and the block-oriented case the synthesis equation $\tilde{\mathbf{x}} = \mathcal{F}\,\mathbf{w}$ can be written as

$$\begin{bmatrix} \vdots \\ \tilde{\mathbf{x}}_l \\ \tilde{\mathbf{x}}_{l+1} \\ \tilde{\mathbf{x}}_{l+2} \\ \vdots \end{bmatrix} = \begin{bmatrix} \ddots & & & \\ & \boxed{\mathbf{F}} & & \\ & & \boxed{\mathbf{F}} & \\ & & & \boxed{\mathbf{F}} \\ & & & & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \mathbf{w}_l \\ \mathbf{w}_{l+1} \\ \mathbf{w}_{l+2} \\ \vdots \end{bmatrix}. \quad (3)$$

Defining the matrices $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L]$, $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_L]$, and $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L]$ the synthesis equation can also be written as $\tilde{\mathbf{X}} = \mathbf{F}\mathbf{W}$. We want to find the frame, $\mathbf{F}$, of size $N \times K$ where $K > N$, and the sparse coefficient vectors, $\mathbf{w}_l$, that minimize the sum of the squared errors. The objective function to be minimized is

$$J = J(\mathbf{F}, \mathbf{W}) = \|\mathbf{X} - \tilde{\mathbf{X}}\|^2 = \|\mathbf{X} - \mathbf{F}\mathbf{W}\|^2. \quad (4)$$

Finding the optimal solution to this problem is difficult if not impossible. We split the problem into two parts to make it more tractable, similar to what is done in the GLA design algorithm for VQ codebooks [11]. The iterative solution strategy presented below results in good, but in general suboptimal, solutions to the problem.

The algorithm starts with a user supplied initial frame $\mathbf{F}^{(0)}$ and then improves it by iteratively repeating two main steps:

1. $\mathbf{W}^{(i)}$ is found by vector selection using frame $\mathbf{F}^{(i)}$. The objective function is $J(\mathbf{W}) = \|\mathbf{X} - \mathbf{F}^{(i)}\mathbf{W}\|^2$, and a sparseness constrain is imposed on $\mathbf{W}$.
2. $\mathbf{F}^{(i+1)}$ is found from $\mathbf{X}$ and $\mathbf{W}^{(i)}$, where the objective function is $J(\mathbf{F}) = \|\mathbf{X} - \mathbf{F}\mathbf{W}^{(i)}\|^2$. This gives:

$$\mathbf{F}^{(i+1)} = \mathbf{X}(\mathbf{W}^{(i)})^T \left(\mathbf{W}^{(i)}(\mathbf{W}^{(i)})^T\right)^{-1} \quad (5)$$

Then we increment i and go to step 1.

i is the iteration number. The first step is suboptimal due to the use of practical vector selection algorithms, while the second step finds the $\mathbf{F}$ that minimizes the objective function.

2.2. Unconstrained Overlapping Frame Design

When we extend our design strategy to the general overlapping case, the large frame, $\mathcal{F}$, can be written as

$$\mathcal{F} = \begin{bmatrix} \ddots & \mathbf{F}_1 & & & \\ \ddots & \vdots & \mathbf{F}_1 & & \\ \ddots & \mathbf{F}_P & \vdots & \mathbf{F}_1 & \ddots \\ & & \mathbf{F}_P & \vdots & \ddots \\ & & & \mathbf{F}_P & \ddots \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \mathbf{F}_1 \\ \vdots \\ \mathbf{F}_P \end{bmatrix}. \quad (6)$$

The synthesis vectors are the columns of $\mathcal{F}$ or $\mathbf{F}$. $\mathbf{F}$ (of size $NP \times K$) can be partitioned into P submatrices, $\{\mathbf{F}_p\}_{p=1}^{P}$ each of size $N \times K$.

We need to rearrange the synthesis equation to be able to use the algorithm from Section 2.1, i.e. extend the algorithm for block based frame design to overlapping frame design. Let $\mathcal{F}$ be defined by (6), and substitute $\mathcal{F}$ into (3). The synthesis of one signal block is now:

$$\tilde{\mathbf{x}}_l = \sum_{p=0}^{P-1} \mathbf{F}_{p+1}\mathbf{w}_{l-p} = \mathbf{F}_1\mathbf{w}_l + \mathbf{F}_2\mathbf{w}_{l-1} + \ldots + \mathbf{F}_P\mathbf{w}_{l-P+1}.$$

A composite synthesis equation for all the signal blocks can be expressed as

$$[\cdots \tilde{\mathbf{x}}_l, \tilde{\mathbf{x}}_{l+1}, \tilde{\mathbf{x}}_{l+2}, \cdots] = [\mathbf{F}_1, \mathbf{F}_2, \cdots, \mathbf{F}_P] \cdot \begin{bmatrix} \cdots & \mathbf{w}_l & \mathbf{w}_{l+1} & \mathbf{w}_{l+2} & \cdots \\ \cdots & \mathbf{w}_{l-1} & \mathbf{w}_l & \mathbf{w}_{l+1} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \mathbf{w}_{l-P+1} & \mathbf{w}_{l-P+2} & \mathbf{w}_{l-P+3} & \cdots \end{bmatrix}. \quad (7)$$

Based on this, with obvious definitions, we establish

$$\tilde{\mathbf{X}} = \widehat{\mathbf{F}}\widehat{\mathbf{W}}. \quad (8)$$

The objective function for step 2 in the design algorithm is now $J(\widehat{\mathbf{F}}) = \|\mathbf{X} - \widehat{\mathbf{F}}\widehat{\mathbf{W}}^{(i)}\|^2$, with the solution

$$\widehat{\mathbf{F}}^{(i+1)} = \mathbf{X}(\widehat{\mathbf{W}}^{(i)})^T \left(\widehat{\mathbf{W}}^{(i)}(\widehat{\mathbf{W}}^{(i)})^T\right)^{-1}. \quad (9)$$

In all other aspects the design algorithm is the same, but we must note that the vector selection step is more involved for the overlapping case than for the block-oriented case [10].

2.3. Constrained Frames — Predefined Structure

Studying (3) and (6) we see that the frames are extensions of block-oriented transforms and (unconstrained) filter banks. Other filter banks, especially the wavelet decomposition, have some predefined structure, such as zero patterns and symmetries in the basis vectors. Imposing similar constrains to the frame we may have: 1) one or more entries are forced to be zero and 2) one or more entries are defined to be equal to another entry or a factor of another entry, i.e. $f(j) = af(i)$ where a usually is +1 or -1, $f(i)$ and $f(j)$ are entries in $\mathbf{F}$. Using these restrictions, the frame (6) may have a structure similar to that of any filter bank or wavelet

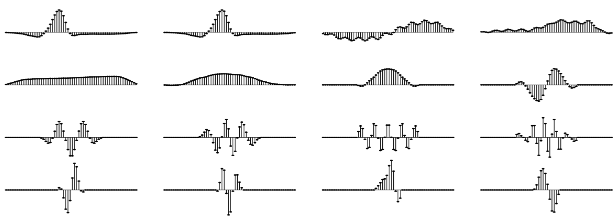

Fig. 1. The 16 synthesis vectors for frame **(e)**.

system. We will now show how the imposed structure affect step 2 in the design algorithm.

Ignoring the superscript index (i) and considering only the overlapping case, which reduces to the block oriented case when $P = 1$, the objective function in step 2 is $J(\widehat{\mathbf{F}}) = \|\mathbf{X} - \widehat{\mathbf{F}}\widehat{\mathbf{W}}\|^2$. This optimizing problem is mathematically the same problem as that of finding a least squares solution to the over-determined set of equations

$$\widehat{\mathbf{W}}^T\widehat{\mathbf{F}}^T = \mathbf{X}^T. \tag{10}$$

Denoting the columns of $\mathbf{X}^T$, i.e. rows of $\mathbf{X}$, as $\{\overline{\mathbf{x}}_n\}_{n=1}^N$, and the columns of $\widehat{\mathbf{F}}^T$, i.e. rows of $\widehat{\mathbf{F}}$, as $\{\overline{\mathbf{f}}_n\}_{n=1}^N$ the equation system can be expanded into

$$\begin{bmatrix} \widehat{\mathbf{W}}^T & & & \\ & \widehat{\mathbf{W}}^T & & \\ & & \ddots & \\ & & & \widehat{\mathbf{W}}^T \end{bmatrix} \begin{bmatrix} \overline{\mathbf{f}}_1 \\ \overline{\mathbf{f}}_2 \\ \vdots \\ \overline{\mathbf{f}}_N \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{x}}_1 \\ \overline{\mathbf{x}}_2 \\ \vdots \\ \overline{\mathbf{x}}_N \end{bmatrix}. \tag{11}$$

With obvious definitions, this can compactly be expressed as $\mathcal{W}\mathbf{f} = \overline{\mathbf{x}}$. The large matrix $\mathcal{W}$ has size $NL \times NKP$. Given the above, we are in a position to precisely explain the implications of the previously mentioned constraints on the problem:

1. If an element of $\mathbf{f}$ is forced to zero, i.e. $f(i) = 0$, this has the consequence of removing variable $f(i)$ in the equation set and deleting one column of $\mathcal{W}$. Thus the problem is formulated in terms of $\tilde{\mathcal{W}}$, which is the same matrix as $\mathcal{W}$, but with column no. i removed.
2. If the relation $f(j) = af(i)$ is imposed on a pair of elements in $\mathbf{f}$, this corresponds to replacing the $\mathcal{W}$ matrix by $\tilde{\mathcal{W}}$ which is found by adding a times column i to column j, and removing column i.

The above operations are repeated a number of times consistent with the number and type of constraints imposed by the frame design specification. If the frame has Q free variables of its total NKP entries, the above operations reduce the number of columns in $\mathcal{W}$ from NKP to Q, the number of rows is unchanged. The solution that find the free variables in the frame is

$$\mathbf{f} = (\mathcal{W}^T\mathcal{W})^{-1}\mathcal{W}^T\overline{\mathbf{x}}. \tag{12}$$

3. SPARSE REPRESENTATION AND COMPRESSION

We will now demonstrate the capabilities of the presented methods when it comes to sparse representation and signal compression. In this work we use the "MIT100" ECG signal which is a normal sinus rhythm. The first 5 minutes (108000 samples) are used for training of the frames and the next five minutes for testing.

Five frames with different structures, denoted **(a)** to **(e)**, were designed. They were compared to three transforms, a block-transform **(f)**, a filter bank **(g)**, and a wavelet **(h)**. The different decomposition methods are now briefly explained, more details on the frames can be found in [10].

- **(a)** Block-oriented frame with size $N = 16$, $K = 32$ and $P = 1$. The number of free variables (all are free) is $Q = NKP = 512$.
- **(b)** Unconstrained overlapping frame with size $N = 16$, $K = 32$ and $P = 2$. The number of free variables (all are free) is $Q = NKP = 1024$.
- **(c)** Unconstrained overlapping frame, with size $N = 8$, $K = 16$ and $P = 4$. The number of free variables (all are free) is $Q = NKP = 512$,
- **(d)** Constrained overlapping frame with size $N = 8$, $K = 16$ and $P = 6$. A structure is imposed on this frame and the number of free variables is $Q = 246$. The constrains are similar to those of frame **(e)**, see below.
- **(e)** Constrained overlapping frame with size $N = 4$, $K = 16$ and $P = 15$. A structure is imposed on this frame and the number of free variables is $Q = 434$. The synthesis vectors are shown in Figure 1, vectors 1-6 are of length $NP = 60$ (1 and 2 are equal except for a translation of 2 samples), vectors 7-12 are of length 28 and forced to be either odd or even symmetric, and frame vectors 13-16 are of length 12.
- **(f)** Discrete Cosine Transform (DCT) with size 32×32, corresponding to a frame where the size is given by $N = 32$, $K = 32$ and $P = 1$.
- **(g)** Lapped Orthogonal Transform (LOT) [12], with size 64×32, corresponding to a frame where the size is given by $N = 32$, $K = 32$ and $P = 2$.
- **(h)** The Daubechies 7-9 bi-orthogonal wavelet filter bank using four levels. A similar reconstruction structure can be imposed by a constrained frame where the size is given by $N = 16$, $K = 16$, and $P = 8$.

The purpose here is to compare the sparse representation efficiencies and the compression capabilities of different frame structures, **(a,b,c,d,e)** to the ones of the transform methods, **(f,g,h)**. A sparse representation is inherent in the frame based representations, as only a limited number of non-zero coefficients are allowed during vector selection. Sparseness is imposed on the other methods by thresholding of the coefficients. The desired sparseness factor gives the number of coefficients to keep; the larger ones are kept and the smaller ones are set to zero. For all decomposition methods the reconstructed signal is formed as a linear combination of the retained synthesis vectors. In the end the signal to noise ratios (SNR) at different sparseness factors are found and compared.

Compression was done using the scheme presented in the introduction. For all decomposition methods above the coefficients (weights) were uniformly quantized with thresholding, the zero bin was twice the size of the other bins. Different bin sizes were used, for each we found the corresponding SNR and the bit rate. For the frame methods the sparseness factor used during vector selection was reduced (more sparseness) as the bin size in the quantizer increased. The quantized coefficients were run-length and entropy

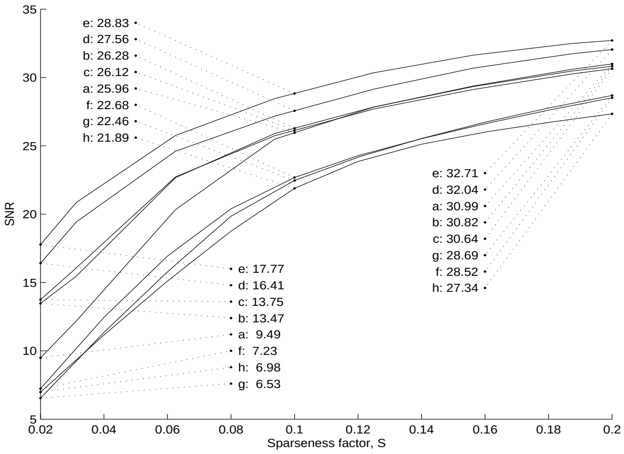

Fig. 2. The achieved Signal to Noise Ratio (SNR) in dB for sparse representation of the test signal for the different frame structures. The sparseness factor S is along the x-axis. The numbers in the figure corresponds to SNR values in dB recorded at that point.

coded, using either a Huffman-coder or an arithmetic coder, depending on what performed best for that frame/method, and the actual bit rate was found. (The Matlab m-files for entropy coding are available at http://www.ux.his.no/˜karlsk/proj99)

The results of the sparse representation experiments are shown in Figure 2. The frames are over-complete, the factor K/N is 2 for frames **(a)** to **(d)** and 4 for frame **(e)**. Thus, it is reasonable to expect the frames to achieve better SNR than the standard decomposition methods for the same sparseness factor. And truly, the frame with the largest ratio K/N has the best SNR for a given sparseness factor. From Figure 2 we see that the frames outperform the transform methods, especially at low sparseness factor, the difference between **(e)** and **(f)** is as much as 10 dB at $S = 0.02$ and more than 6 dB at $S = 0.1$.

For compression it is not obvious that the frames will be superior to the transform methods. The over-complete frames demand more position information identifying the non-zero coefficients. In addition, using frames the location of the non-zero weights is more spread out than using a common method like DCT where the coefficients within each block are sorted by frequency. Note that the sparseness factor was small in the compression experiments, especially for the (best) frames, for frame **(e)** it was in range $0.02 \leq S \leq 0.08$.

In Figure 3 the achieved bit rate (bit per sample) is along the x-axis. Here we note that the block-oriented frame **(a)** is not good. The main reason is probably the relatively short synthesis vectors of length 16. A frame with size 32×64 would be expected to perform better, and would be a more natural comparison to the 32×32 DCT. The transform methods and frames **(b)** and **(c)** are all quite close in performance, the DCT is a little bit better than the rest. But the best results are achieved by the frames **(d)** and **(e)** and we note that frame **(e)** keeps the lead we observed in the sparse representation experiments, see Figure 2. At bit rates lower than 0.45 its performing advantage relative to DCT is more than 2 dB.

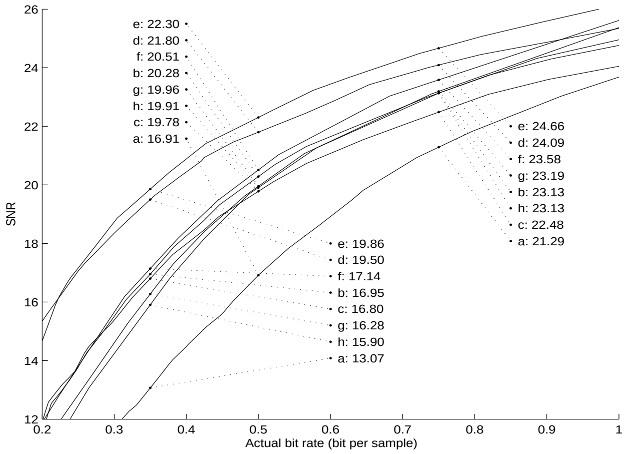

Fig. 3. The achieved SNR for compression at different bit rates.

4. CONCLUSIONS

This paper has presented a flexible method for the design of signal dependent frames. The sparse representation capabilities and the compression performance at low bit rates of the designed frames were shown to be excellent.

5. REFERENCES

[1] Z. Lu, D. Y. Kim, and W. A. Pearlman, "Wavelet compression of ECG signals by the set partitioning in hierarchical trees algorithm," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 7, pp. 849–856, July 2000.

[2] Y. Zigel, A. Cohen, and A. Katz, "ECG signal compression using analysis by synthesis coding," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 10, pp. 1308–1316, Oct. 2000.

[3] R. Nygaard, G. Melnikov, and A. K. Katsaggelos, "A rate distortion optimal ECG coding algorithm," *IEEE Trans. Biomed. Eng.*, vol. 48, no. 1, pp. 28–40, Jan. 2001.

[4] Massachusetts Institute of Technology, *The MIT-BIH Arrhythmia Database CD-ROM*, MIT, 2 edition, 1992.

[5] S. Mallat and F. Falzon, "Analysis of low bit rate image transform coding," *IEEE Trans. Signal Processing*, vol. 46, pp. 1027–1042, Apr. 1998.

[6] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM journal on computing*, vol. 24, pp. 227–234, Apr. 1995.

[7] S. F. Cotter, J. Adler, B. D. Rao, and K. Kreutz-Delgado, "Forward sequential algorithms for best basis selection," *IEE Proc. Vis. Image Signal Process*, vol. 146, no. 5, pp. 235–244, Oct. 1999.

[8] K. Skretting and J. H. Husøy, "Partial search vector selection for sparse signal representation," in *NORSIG-03*, Bergen, Norway, Oct. 2003, Also available at http://www.ux.his.no/˜karlsk/.

[9] K. Engan, *Frame Based Signal Representation and Compression*, Ph.D. thesis, Norges teknisk-naturvitenskapelige universitet (NTNU)/Høgskolen i Stavanger, Sept. 2000, Available at http://www.ux.his.no/˜kjersti/.

[10] K. Skretting, *Sparse Signal Representation using Overlapping Frames*, Ph.D. thesis, NTNU Trondheim and Høgskolen i Stavanger, Oct. 2002, Available at http://www.ux.his.no/˜karlsk/.

[11] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Norwell, Mass., USA, 1992.

[12] H. S. Malvar and D. H. Staelin, "The LOT: Transform coding without blocking effects," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 4, pp. 553–559, Apr. 1989.