Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Signal Processing 86 (2006) 117–126

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

General design algorithm for sparse frame expansions ☆

Karl Skretting*, John Håkon Husøy, Sven Ole Aase

Department of Electrical and Computer Engineering, University of Stavanger, PO Box 2557, Ullandhaug, N-4036 Stavanger, Norway

Received 5 May 2003; received in revised form 17 December 2004; accepted 24 April 2005
Available online 2 June 2005

Abstract

Signal expansions using frames may be considered as generalizations of signal representations based on transforms and filter banks. Frames, or dictionaries, for sparse signal representations may be designed using an iterative algorithm with two main steps: (1) Frame vector selection and expansion coefficient determination for signals in a *training set*, selected to be representative of the signals for which compact representations are desired, using the frame designed in the previous iteration. (2) Update of frame vectors with the objective of improving the representation of step (1). This method for frame design was used by [Engan et al., Signal Processing 80 (2000) 2121–2140] for block-oriented signal expansions, i.e. generalizations of block-oriented transforms and by [Aase et al., IEEE Trans. Signal Process. 49(5) (2001) 1087–1096] for non-block-oriented frames—for short *overlapping frames*, that may be viewed as generalizations of critically sampled filter banks. Here we give the solution to the *general frame design problem* using the compact notation of linear algebra. This makes the solution both conceptually and computationally easier, especially for the overlapping frame case. Also, the solution is more general than those presented earlier, facilitating the imposition of constraints, such as symmetry, on the designed frame vectors.

Keywords: Frame; Over-complete; Dictionary; Frame design; Matching pursuit; Filter banks; Sparse signal representation

1. Introduction

Sparse representations of signals can be constructed using an over-complete dictionary, i.e. a frame, and a Matching Pursuit algorithm [1–5]. In recent years this has found several applications: Low bit rate video coding [6–10], image compression [11–15], and others [16–18]. Different kinds of frames have been used. Examples include the frame created by concatenation of the orthogonal basis vectors of the DCT (Discrete Cosine Transform) and those of the Haar transform [13], Gabor functions [6,8], (oversampled) filter banks and wavelet trees [10], and Gaussian chirps [19]. These selections are motivated by their good time-frequency resolution and effective implementation. The design of frames, adapted to a class of signals,

☆Supported by Center for Information Technology (SIKT), University of Stavanger.
*Corresponding author. Tel.: +47 51832016; fax: +47 51831750.
E-mail addresses: karl.skretting@uis.no (K. Skretting), john.h.husoy@uis.no (J.H. Husøy), sven.o.aase@uis.no (S.O. Aase).

0165-1684/$ - see front matter
doi:10.1016/j.sigpro.2005.04.013

defined through the availability of a *training set* of signals, has not been given much attention. Some of the available works are [1,2,20–22]. Here we present such a theory, encompassing [1,2] as special cases. A key element in this work is that by rearranging or reformulating the problem in appropriate ways, we are able to apply standard linear algebra methods in the solution. The frame design method as presented here is not only related to block-oriented transforms but it is extended to frames that have a structure similar to filter banks (overlapping frames) and wavelets (general frames, or constrained frames with predefined structure).

An over-complete set of N-dimensional vectors, spanning the space $\mathbb{R}^N$, $\{\mathbf{f}_k\}_{k=1}^K$ where $K \geqslant N$, is a *frame*. The frame concept was first introduced in the early fifties by Duffin and Schaeffer [23]. In the late eighties frames received renewed attention, mainly as a consequence of identified connections with the wavelet transform and time-frequency analysis [24,25]. In this paper frames are represented as follows: A frame is given by a matrix $\mathbf{F}$ of size $N \times K$, $K \geqslant N$, where the columns are the frame vectors, $\mathbf{f}_k$. A signal block, $\mathbf{x}_l$, can be represented by a weighted sum of these frame vectors

$$\tilde{\mathbf{x}}_l = \sum_{k=1}^{K} w_l(k)\mathbf{f}_k = \mathbf{F}\mathbf{w}_l. \tag{1}$$

This is a *signal expansion* that, depending on the selection of weights, $\mathbf{w}_l$, may be an exact or approximate representation of the signal block. Since we are representing blocks of a signal, the frames in question are referred to as *block-oriented*. In this paper we focus on approximate *sparse* representations in which a small number of the weights $w_l(k)$ are non-zero. The weights, $w_l(k)$, can be represented by column vectors, $\mathbf{w}_l$, each with K elements. $\mathbf{F}$ represents a frame of the space $\mathbb{R}^N$, assuming that the frame vectors span $\mathbb{R}^N$.

In a *sparse representation* most of the weights of the signal expansion of Eq. (1) are zero. To quantify the degree of sparseness we define a *sparseness factor* given by the proportion of non-zero weights in the signal expansion to the number of samples in the signal:

$$S = \frac{\text{Average number of non-zero weights in } \mathbf{w}_l}{\text{Number of signal samples in } \mathbf{x}_l}. \tag{2}$$

It is convenient to interpret Eq. (1) as a *synthesis equation* since it explicitly indicates how the signal approximation is synthesized. In line with this, the frame vectors are called *synthesis vectors*. This synthesis is the same operation as the reconstruction step in a transform coder [26]. The block-oriented frame representation can thus be considered a generalization of the block-oriented transform representation in the sense that more synthesis vectors are available when building the reconstructed signal. This is a consequence of selecting $K \geqslant N$. Similarly, as we will see in Section 3, the *overlapping frame* can be considered a generalization of a critically sampled synthesis filter bank. A more complete presentation of the use of frames for sparse representation, and the relation of the frame concept to block transforms, critically sampled filter banks as well as wavelets can be found in [1] and [2].

In the following section we introduce notation used throughout the paper and precisely formulate the frame design problem. For presentation purposes this is done for the block-oriented case, and the solution for this case is presented. Following this, in the main part of the paper—Section 3, we first introduce the overlapping frame, then give definitions of two auxiliary matrices leading up to a formulation of the overlapping frame design problem in a form enabling us to directly make use of the solution derived for the block-oriented case. We proceed in Section 4 by showing how constraints on the synthesis frame vectors can be incorporated into the problem formulation. Finally, we present some design examples and illustrate the sparse representation capabilities of the designed frames.

2. Block-oriented frame design

The frame design methodology presented here was first used in the context of block-oriented frame design [1]. The frame should be adapted to a class of signals, represented by a large set of training vectors, $\{\mathbf{x}_l\}_{l=1}^{L}$, in a way that makes the frame well suited for a sparse representation for this class of signals. For our frame design algorithm it is convenient to collect the training vectors, which are consecutive blocks of a training signal $\mathbf{x}$, the synthesized vectors, the weight vectors and the frame vectors into matrices,

$$\begin{aligned} \mathbf{X} &= [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \mathbf{x}_3 \quad \ldots \quad \mathbf{x}_L], \\ \tilde{\mathbf{X}} &= [\tilde{\mathbf{x}}_1 \quad \tilde{\mathbf{x}}_2 \quad \tilde{\mathbf{x}}_3 \quad \ldots \quad \tilde{\mathbf{x}}_L], \\ \mathbf{W} &= [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \mathbf{w}_3 \quad \ldots \quad \mathbf{w}_L], \\ \mathbf{F} &= [\mathbf{f}_1 \quad \mathbf{f}_2 \quad \ldots \quad \mathbf{f}_K]. \end{aligned} \tag{3}$$

The synthesis equation, Eq. (1), may now be written as

$$\tilde{\mathbf{X}} = \mathbf{F}\mathbf{W}. \tag{4}$$

An equivalent way to write the synthesis equation is $\tilde{\mathbf{x}} = \mathscr{F}\mathbf{w}$, which can be illustrated as follows:

$$\begin{bmatrix} \vdots \\ \tilde{\mathbf{x}}_l \\ \tilde{\mathbf{x}}_{l+1} \\ \tilde{\mathbf{x}}_{l+2} \\ \vdots \end{bmatrix} = \begin{bmatrix} \ddots & & & \\ & \boxed{\mathbf{F}} & & \\ & & \boxed{\mathbf{F}} & \\ & & & \boxed{\mathbf{F}} \\ & & & & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \mathbf{w}_l \\ \mathbf{w}_{l+1} \\ \mathbf{w}_{l+2} \\ \vdots \end{bmatrix}. \tag{5}$$

This latter equation clearly reveals the block-oriented structure in the synthesis equation, the large matrix $\mathscr{F}$ is a block-diagonal matrix.

Frame design, or the problem of seeking the optimal frame for a given class of signals and a given sparseness factor, is briefly summarized below. More details can be found in [1]. The objective is to find the frame, $\mathbf{F}$, that minimizes the approximation error $\|\mathbf{x} - \tilde{\mathbf{x}}\|$. For the block-oriented case this can be expressed as an optimization problem $\min_{\mathbf{F},\mathbf{W}} J(\mathbf{F}, \mathbf{W})$ where

$$J(\mathbf{F}, \mathbf{W}) = \|\mathbf{x} - \tilde{\mathbf{x}}\|^2 = \|\mathbf{X} - \mathbf{F}\mathbf{W}\|^2, \tag{6}$$

subject to a sparsity constraint on $\mathbf{W}$. The norm used is $\|\mathbf{A}\|^2 = \mathrm{trace}(\mathbf{A}^{\mathrm{T}}\mathbf{A})$. This norm is used both for matrices and vectors, for vectors it corresponds to the ordinary 2-norm. Finding the optimal solution to this problem is difficult if not impossible. A practical optimization strategy, not necessarily leading to a global optimum, but with established good performance [1,2], can be found by splitting the problem into two parts which are alternately solved within an iterative loop. The method is inspired by the generalized Lloyd algorithm [27] and can be interpreted as a generalization of this algorithm.

The approach starts with a user-supplied initial frame $\mathbf{F}^{(0)}$ and then proceeds to improve it by iteratively repeating two main steps using the training signals. The ith iteration can be described as:

(1) $\mathbf{W}^{(i)}$ is found by vector selection and weight computation based on the frame $\mathbf{F}^{(i)}$, where the objective function is $J(\mathbf{W}) = \|\mathbf{X} - \mathbf{F}^{(i)}\mathbf{W}\|^2$ and a sparseness constraint is imposed on $\mathbf{W}$. This problem is known to be NP-hard [4,28]. Nevertheless several practical approaches employing *matching pursuit* algorithms [29] are known to work well in the vector selection and weight computation. In the present work we employ an order recursive matching pursuit algorithm described in [30] for this purpose.

(2) $\mathbf{F}^{(i+1)}$ is found from $\mathbf{X}$ and $\mathbf{W}^{(i)}$, where the objective function, for the block-oriented case, is $J(\mathbf{F}) = \|\mathbf{X} - \mathbf{F}\mathbf{W}^{(i)}\|^2$. In general the objective function is $J(\mathbf{F}) = \|\mathbf{x} - \tilde{\mathbf{x}}\|^2$ and purposeful manipulations are needed to solve this, as will be shown in Sections 3 and 4. Here though, it is straightforward. Transposing the objective function gives $J(\mathbf{F}) = \|\mathbf{X}^{\mathrm{T}} - (\mathbf{W}^{(i)})^{\mathrm{T}}\mathbf{F}^{\mathrm{T}}\|^2$. This optimization problem is mathematically the same problem as that of finding a least squares solution to an over-determined set of linear equations [31]. The solution is $\mathbf{F}^{\mathrm{T}} = (\mathbf{W}\mathbf{W}^{\mathrm{T}})^{-1}\mathbf{W}\mathbf{X}^{\mathrm{T}}$, which for the present case gives:

$$\mathbf{F}^{(i+1)} = \mathbf{X}(\mathbf{W}^{(i)})^{\mathrm{T}}(\mathbf{W}^{(i)}(\mathbf{W}^{(i)})^{\mathrm{T}})^{-1}. \tag{7}$$

Then we increment i and go to step 1 above unless some stopping criterion is satisfied.

Using this algorithm the resulting frame will depend on (1) the selected structure of $\mathbf{F}$, giving its size and the number of free variables, (2) the initial values chosen for the frame $\mathbf{F}^{(0)}$, (3) the training signal $\mathbf{x}$ representing the signal class, and (4) the target sparseness factor used in vector selection in step 1 above. From Eq. (4) and Eq. (7) the reconstructed signal is $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{W}^{\mathrm{T}}(\mathbf{W}\mathbf{W}^{\mathrm{T}})^{-1}\mathbf{W}$. Here, $\mathbf{W}^{\mathrm{T}}(\mathbf{W}\mathbf{W}^{\mathrm{T}})^{-1}\mathbf{W}$ is a projection matrix, thus each row of $\tilde{\mathbf{X}}$ is formed by projecting the corresponding row of $\mathbf{X}$ onto the space spanned by the rows of $\mathbf{W}$. The solution in Eq. (7) assumes that $\mathbf{W}^{\mathrm{T}}$ has full rank, else the inverse of $(\mathbf{W}\mathbf{W}^{\mathrm{T}})$ does not exist. Experiments done have shown that the full rank assumption is usually met. When it is not, the case is usually that one of the synthesis vectors of $\mathbf{F}$ is not used for the sparse representation of *any* of the training vectors (this might happen if the set of training vectors is too small, normally it should be at least $L > 5K$), then a row of $\mathbf{W}$ will consist of zeros only and $\mathbf{W}^{\mathrm{T}}$ will be rank deficient. If this happens the solution can be found by removing the unused synthesis vector from $\mathbf{F}$ and the zero row from $\mathbf{W}$ and solve the equation system for the rest of the frame vectors. The unused frame vector may be replaced by a random vector, or a segment of the training signal, this will hopefully cause it to be used during the vector selection step in the next iteration.

3. Overlapping frame design

In many signal processing applications critically sampled filter banks are known to perform better than block-oriented transforms. A uniform synthesis K channel filter bank is shown in Fig. 1. Using the input–output relations for an up-sampler and a linear filter [32], the input-output relation for the synthesis filter bank can be verified to correspond

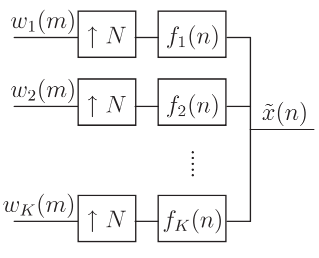

Fig. 1. A uniform synthesis filter bank of K filters, where the upsampling factor is N for each filter. When $K > N$ the filter bank is not critically sampled. The filters are assumed to be of the same length, PN.

to the synthesis equation $\tilde{\mathbf{x}} = \mathscr{F}\mathbf{w}$ with $\mathscr{F}$ being a band-diagonal matrix:

$$\mathscr{F} = \begin{bmatrix} \ddots & \mathbf{F}_0 & & & \\ \ddots & \vdots & \mathbf{F}_0 & & \\ \ddots & \mathbf{F}_{P-1} & \vdots & \mathbf{F}_0 & \ddots \\ & & \mathbf{F}_{P-1} & \vdots & \ddots \\ & & & \mathbf{F}_{P-1} & \ddots \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \mathbf{F}_0 \\ \vdots \\ \mathbf{F}_{P-1} \end{bmatrix}. \tag{8}$$

The K synthesis vectors, each of length NP, are the columns of $\mathbf{F}$, size $NP \times K$. $\mathbf{F}$ is partitioned into P sub-matrices, $\{\mathbf{F}_p\}_{p=0}^{P-1}$ each of size $N \times K$. At each repetition of $\mathbf{F}$ in $\mathscr{F}$, $\mathbf{F}$ is moved N positions down and K positions to the right. The difference between a synthesis block transform and a synthesis filter bank can be seen quite easily by comparing Eqs. (5) and (8). Having $\mathbf{F}$ as a square, invertible matrix, i.e. $K = N$, in Eq. (5) we get the synthesis equation of a block-oriented transform, while having $\mathscr{F}$ as in Eq. (8) and $K = N$ we get the critically sampled synthesis filter bank. In both cases the synthesis equation can be written as $\tilde{\mathbf{x}} = \mathscr{F}\mathbf{w}$, and $\mathscr{F}$ is a large, square, usually invertible, matrix. For the filter bank case special considerations should be taken at the signal ends. In both cases we get the extension to a frame when $K > N$, a block-oriented frame for the block-oriented transform ($P = 1$), and a overlapping frame for the filter bank case ($P > 1$). We refer to P as the overlap factor.

The term "overlapping frame" comes from the fact that the synthesis vectors of neighboring blocks in Eq. (8) overlap each other. The overlapping frame is not a new concept, it is exactly the same as an *oversampled* synthesis filter bank with K filters and up-sampling factor N [33,34]. Readers familiar with filter bank theory can note that the splitting of $\mathbf{F}$ into P sub-matrices corresponds to the polyphase representation of the synthesis filter bank. In fact, the polyphase matrix is $\mathbf{R}(z) = \sum_{p=0}^{P-1} \mathbf{F}_p z^{-p}$.

We need to rearrange the synthesis equation to be able to use the algorithm from Section 2, i.e. extend the algorithm for block based frame design to overlapping frame design. Let $\mathscr{F}$ be defined by Eq. (8), and substitute $\mathscr{F}$ into Eq. (5). The synthesis equation for a signal block can now be written in terms of the sub-matrices of $\mathbf{F}$:

$$\begin{aligned}\tilde{\mathbf{x}}_l &= \sum_{p=0}^{P-1} \mathbf{F}_p \mathbf{w}_{l-p} \\ &= \mathbf{F}_0\mathbf{w}_l + \mathbf{F}_1\mathbf{w}_{l-1} + \cdots + \mathbf{F}_{P-1}\mathbf{w}_{l-P+1} \\ &= [\mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_{P-1}] \begin{bmatrix} \mathbf{w}_l \\ \mathbf{w}_{l-1} \\ \vdots \\ \mathbf{w}_{l-P+1} \end{bmatrix}. \end{aligned} \tag{9}$$

Defining

$$\widehat{\mathbf{F}} = [\mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_{P-1}] \quad \text{and} \tag{10}$$

$$\widehat{\mathbf{W}} = \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_l & \cdots & \mathbf{w}_L \\ \mathbf{w}_0 & \cdots & \mathbf{w}_{l-1} & \cdots & \mathbf{w}_{L-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{w}_{-P+1} & \cdots & \mathbf{w}_{l-P+1} & \cdots & \mathbf{w}_{L-P+1} \end{bmatrix} \tag{11}$$

the synthesis equation for all the signal blocks, 1 through L, can be written as

$$\tilde{\mathbf{X}} = \widehat{\mathbf{F}}\widehat{\mathbf{W}}. \tag{12}$$

The synthesis equation in Eq. (12) is in a form suitable for our purpose, it gives the objective function for step 2 in the design algorithm as $J(\widehat{\mathbf{F}}) = \|\mathbf{X} - \widehat{\mathbf{F}}\widehat{\mathbf{W}}^{(i)}\|^2$. The solution—in complete analogy with the block-oriented case, is

$$\widehat{\mathbf{F}}^{(i+1)} = \mathbf{X}(\widehat{\mathbf{W}}^{(i)})^{\mathrm{T}}(\widehat{\mathbf{W}}^{(i)}(\widehat{\mathbf{W}}^{(i)})^{\mathrm{T}})^{-1}. \tag{13}$$

In all other aspects the design algorithm is the same, but note that the vector selection step is more involved for the overlapping case than for the block-oriented case [35].

Note that the exact contents of the first few columns of $\widehat{\mathbf{W}}$, i.e. those columns $\mathbf{w}_j$ with $j<1$, depends on assumptions on the signal outside the range given by $\{\mathbf{x}_l\}_{l=1}^{L}$. In practice L, the number of blocks in the training signal set, is in the order of several thousands, implying that whatever assumptions are made on the signal outside the training set is of minor importance. Nevertheless, one simple way to solve the end-effect-problem is to assume circular extension of both the signal and the weights, $\mathbf{x}_j = \mathbf{x}_{L+j}$ and $\mathbf{w}_j = \mathbf{w}_{L+j}$ for all j. With this the $\mathscr{F}$ matrix, of size $NL \times NK$, for the case where $P = 3$ becomes

$$\mathscr{F} = \begin{bmatrix} \mathbf{F}_0 & & & & \mathbf{F}_2 & \mathbf{F}_1 \\ \mathbf{F}_1 & \mathbf{F}_0 & & & & \mathbf{F}_2 \\ \mathbf{F}_2 & \mathbf{F}_1 & \ddots & & & \\ & \mathbf{F}_2 & \ddots & \mathbf{F}_0 & & \\ & & \ddots & \mathbf{F}_1 & \mathbf{F}_0 & \\ & & & \mathbf{F}_2 & \mathbf{F}_1 & \mathbf{F}_0 \end{bmatrix}. \tag{14}$$

4. Constrained overlapping frames

As pointed out previously, the frame vectors of Section 3 correspond to the filter unit pulse responses of the synthesis filter bank of Fig. 1. This filter bank is restricted in the sense that all channels have the same upsampling factor, and the channel filters are all of the same length (NP). A more general synthesis filter bank allowing different upsampling ratios and different filter lengths is illustrated in Fig. 2. This filter bank has J different FIR channel filters. The length of the filter for channel j is denoted by l_j and the upsampling factor by n_j. As detailed in [2] this structure encompasses every conceivable transform, filter bank, and wavelet decomposition expansion along with their generalizations.

To make a frame corresponding to this general filter bank structure we can proceed as outlined in

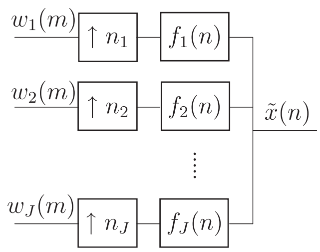

Fig. 2. A general synthesis filter bank of J filters.The filter length, l_j, and upsampling factor, n_j, may vary for each filter.

the beginning of Section 3, by formulating the input–output relations of the various channels and collecting them using appropriately defined matrix/vector quantities. Carrying out this quite laborious, but straightforward, task, we find that the structure of the synthesis equation, $\tilde{\mathbf{x}} = \mathscr{F}\mathbf{w}$, can be maintained [2].

In general, the following modifications to the quantities N, K and P, that collectively determine the structure of $\mathscr{F}$ in Eq. (8), must be kept in mind:

$$N = \text{least common multiple of } \{n_j\}_{j=1}^{J},$$
$$K = \sum_{j=1}^{J} \frac{N}{n_j},$$
$$P = \max_j \left\lceil \frac{l_j - n_j}{N} \right\rceil + 1, \tag{15}$$

where $\lceil x \rceil$ is the smallest integer larger or equal to x. Depending on the desired lengths, l_j, and the upsampling factors, n_j, the $\mathbf{F}$ matrix will be populated by a combination of elements of the frame vectors and zeros. Those zeros can be interpreted as constraints on the $\mathbf{F}$ matrix. Obviously these constraints must be embedded into the frame design algorithm.

In many design problems in signal processing, the imposition of various symmetries plays an important role. For example, in the design of filters for critically sampled filter banks we may desire filters with linear phase, i.e. unit pulse responses that are symmetric or antisymmetric with respect to their midpoints. Such symmetries can be imposed by expressing relations between pairs of elements of frame vectors of type $f(i) = af(j)$, where we have assumed that the elements of all frame vectors are indexed sequentially. Most often a will be given by 1 (to specify even symmetries) or -1 (to specify odd symmetries). In the following we reformulate the design problem presented previously in such a way as to facilitate the incorporation of the two types of constraints described above.

Recall that what we have done in Section 3, is to pose the problem as that of finding a least squares solution to an overdetermined set of linear equations. This is our goal here too. Transposing Eq. (12) we get

$$\widehat{\mathbf{W}}^{\mathrm{T}}\widehat{\mathbf{F}}^{\mathrm{T}} = \mathbf{X}^{\mathrm{T}}. \tag{16}$$

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

$$\begin{bmatrix} \widehat{\mathbf{W}}^{\mathrm{T}} & & & \\ & \widehat{\mathbf{W}}^{\mathrm{T}} & & \\ & & \ddots & \\ & & & \widehat{\mathbf{W}}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \overline{\mathbf{f}}_1 \\ \overline{\mathbf{f}}_2 \\ \vdots \\ \overline{\mathbf{f}}_N \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{x}}_1 \\ \overline{\mathbf{x}}_2 \\ \vdots \\ \overline{\mathbf{x}}_N \end{bmatrix}. \tag{17}$$

With obvious definitions, this can compactly be expressed as

$$\mathscr{W}\mathbf{f} = \overline{\mathbf{x}}. \tag{18}$$

The large matrix $\mathscr{W}$ 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 $\mathscr{W}$. Thus the problem is formulated in terms of $\tilde{\mathscr{W}}$, which is the same matrix as $\mathscr{W}$, 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 $\mathscr{W}$ matrix by $\tilde{\mathscr{W}}$ 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 $\mathscr{W}$ from NKP to Q, the number of rows is unchanged. The solution that gives the free variables in the frame is

$$\mathbf{f} = (\mathscr{W}^{\mathrm{T}}\mathscr{W})^{-1}\mathscr{W}^{\mathrm{T}}\overline{\mathbf{x}}. \tag{19}$$

5. Sparse representation examples

We will now demonstrate the representation capabilities of the presented algorithms. In this work we use the "MIT100" ECG signal which is a normal sinus rhythm [36], Fig. 3. As the heart beats are triggered by impulses with a frequency in range 1–2 Hz, we may expect that a sparse representation is relevant for the ECG signal which here is sampled at 360 Hz. The first 5 min (108000 samples) are used for training of the frames and the next 5 min for testing.

Four frames with different structures, denoted **(a)**–**(d)** were designed. Their performances were compared to three common methods, a block-oriented transform **(e)**, a filter bank **(f)**, and a wavelet **(g)**. The different decomposition methods are now explained.

(a) Block-oriented frame with size $N = 32$, $K = 64$ and $P = 1$. The number of free variables (all are free) is $Q = NKP = 2048$.

(b) Unconstrained overlapping frame with size $N = 16$, $K = 32$ and $P = 4$. The number of free variables (all are free) is $Q = NKP = 2048$.

(c) Constrained overlapping frame where the imposed structure is given, referring to Fig. 2, by $J = 10$ synthesis filters, filter f_j has length l_j and upsampling factor n_j as given by the table below. Also symmetry constraints are imposed: '-' for none, 'o' for odd, and 'e' for even.

j:	1	2-4	5-7	8-10
l_j:	58	60	24	24
n_j:	2	4	8	8
Sym.:	-	-	o	e

The structure of **F** is given by Eq. (15) which gives $N = 8$, $K = 16$ and $P = 8$ and the number of free variables is $Q = 58 + 3 \cdot 60 + 6 \cdot 12 = 310$.

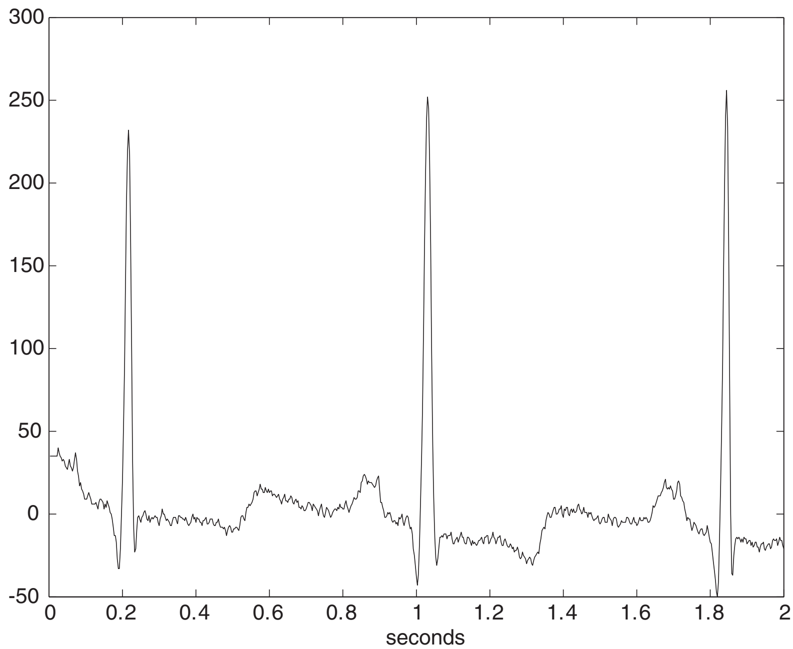

Fig. 3. The first 2 s (720 samples) of the electrocardiogram (ECG) training signal. Three QRS complexes are shown, at 0.2, 1 and 1.8 s.

(d) Constrained overlapping frame where the imposed structure is given, referring to Fig. 2, by $J = 15$ synthesis filters specified by

j:	1-2	3-4	5-8	9-12	13	14	15
l_j:	74	76	32	32	48	10	10
n_j:	2	4	8	8	2	2	2
Sym.:	-	-	o	e	-	e	o

The structure of $\mathbf{F}$ is now given by Eq. (15), giving $N = 8$, $K = 32$ and $P = 10$, and the number of free variables is $Q = 2 \cdot 74 + 2 \cdot 76 + 8 \cdot 16 + 48 + 2 \cdot 5 = 486$. The synthesis filters are shown in Fig. 4.

(e) Discrete Cosine Transform (DCT) with size 32×32, corresponding to a frame where the size is given by $N = 32$, $K = 32$ and $P = 1$.

(f) Lapped Orthogonal Transform (LOT) [37], with size 64×32, corresponding to a frame where the size is given by $N = 32$, $K = 32$ and $P = 2$.

(g) The Daubechies 7–9 biorthogonal wavelet filter bank using five levels. A similar reconstruction structure can be imposed by a constrained frame where the size is given by $N = 32$, $K = 32$, and $P = 8$.

The purpose here is to compare the sparse representation capabilities for different frame structures, (**a,b,c,d**) to the ones of the common methods, (**e, f, g**). A sparse representation is inherent for 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,

$$\mathrm{SNR} = 20 \log_{10} \frac{\|\mathbf{x}\|}{\|\mathbf{x} - \tilde{\mathbf{x}}\|} \text{ [dB]}, \tag{20}$$

at different sparseness factors are found and compared. The results of the sparse representation experiments are shown in Fig. 5. The frames are over-complete, the factor K/N is 2 for frames (**a**)–(**c**) and 4 for frame (**d**). 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 Fig. 5 we see that the frames outperform the

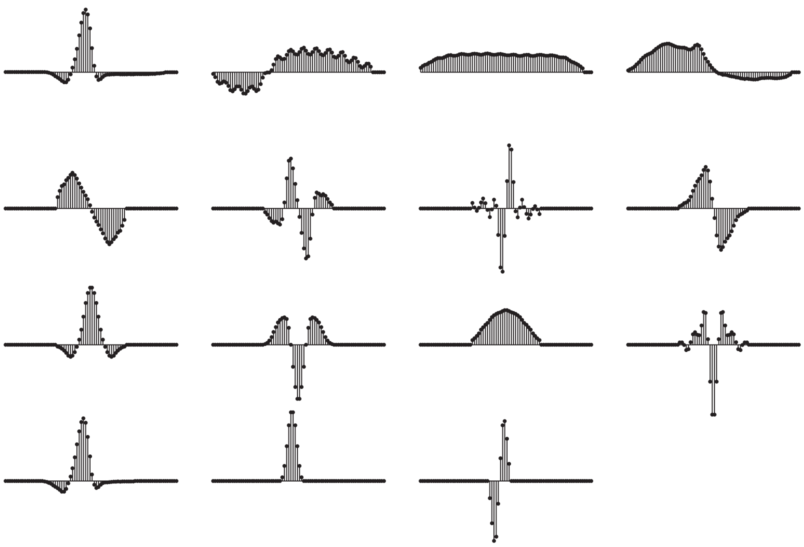

Fig. 4. The 15 synthesis filters for frame (**d**). Note that the symmetry constrains are fulfilled, filters f_5–f_8 (second row) and f_{15} are odd symmetric and filters f_9–f_{12} (third row) and f_{14} are even symmetric.

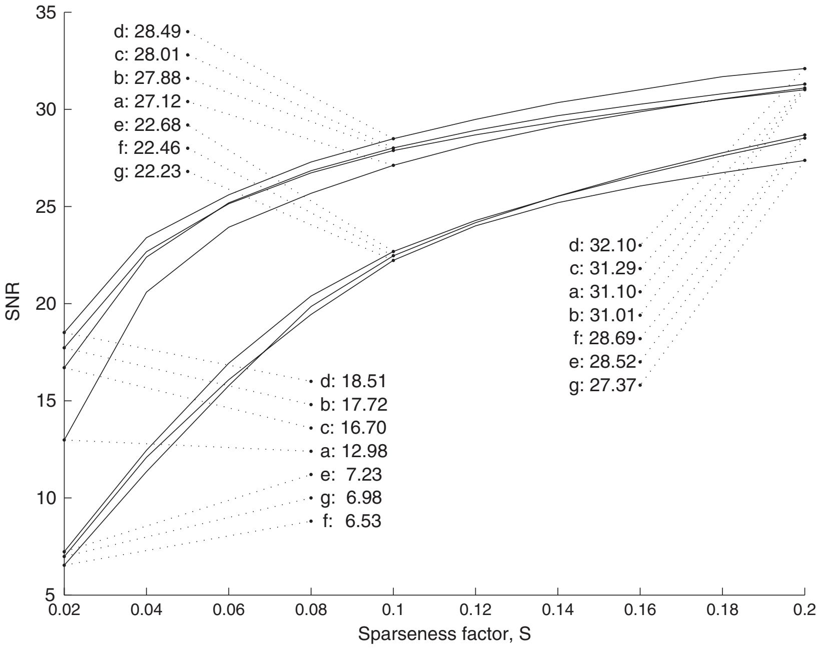

Fig. 5. The achieved signal-to-noise ratio (SNR) in dB for sparse representation of the test signal for the different frame structures. The sparseness factor S is along the x-axis. The numbers in the figure corresponds to SNR values in dB recorded at that point.

transform methods, especially at low sparseness factors, the difference between (**d**) and (**e**) is as much as 10 dB at $S = 0.02$ and almost 6 dB at $S = 0.1$. For this signal, and at low sparseness factors, we would expect that longer synthesis vectors are better than short ones. This seems to be true only when the synthesis vectors are adapted to the signal, the method (**e**) (longest synthesis filter has length 32) is better than both method (**f**) (longest synthesis filter has length 64) and method (**g**) (longest synthesis filter has length 249) even when the sparseness factor is as small as 0.02. For the frames though, longer synthesis vectors seem to be advantageous. The training of the frames captures shapes from the training signal and uses them in the synthesis vectors, in Fig. 4 we recognize segments from the ECG signal in Fig. 3, i.e. the QRS complex in f_1 and f_{13}.

6. Conclusion

In this article, we derived general solutions to the “find the frame given the weights”-step used in the frame design method in [1] and [2]. These derivations are conceptually easier, more compactly expressed, and the solutions are more general than those presented previously. The derived solutions also facilitate the inclusion of various design constraints. It was shown that the designed frames adapt well to the training signal, and that this gives excellent sparse representations of signals belonging to the same class.

References

[1] K. Engan, S.O. Aase, J.H. Husøy, Multi-frame compression: theory and design, Signal Processing 80 (2000) 2121–2140.

[2] S.O. Aase, J.H. Husøy, K. Skretting, K. Engan, Optimized signal expansions for sparse representation, IEEE Trans. Signal Process. 49 (5) (2001) 1087–1096.

[3] S.G. Mallat, Z. Zhang, Matching pursuit with time-frequency dictionaries, IEEE Trans. Signal Process. 41 (12) (1993) 3397–3415.

[4] G. Davis, Adaptive nonlinear approximations, Ph.D. Thesis, New York University, September 1994.

[5] B.D. Rao, Signal processing with the sparseness constraint, in: Proceedings of the ICASSP '98, Seattle, USA, 1998, pp. 1861–1864.

[6] R. Neff, A. Zakhor, Very low bit-rate video coding based on matching pursuit, IEEE Trans. Circuits Syst. Video Technol. 7 (1) (1997) 158–171.

[7] R. Neff, A. Zakhor, Modulus quantization for matching-pursuit video coding, IEEE Trans. Circuits Syst. Video Technol. 10 (6) (2000) 895–912.

[8] M.R. Banham, J.C. Brailean, A selective update approach to matching pursuits video coding, IEEE Trans. Circuits Syst. Video Technol. 7 (1) (1997) 119–129.

[9] M. Vetterli, T. Kalker, Matching pursuit for compression and application to motion compensated video coding, in: Proceedings of the IEEE International Conference on Image Processing, ICIP-94, vol. 1, 1994, pp. 725–729.

[10] C.D. Vleeschouwer, B. Macq, Subband dictionaries for low-cost matching pursuits of video residues, IEEE Trans. Circuits Syst. Video Technol. 9 (7) (1999) 984–993.

[11] F. Bergeaud, S. Mallat, Matching pursuit of images, in: Proceedings of the IEEE International Conference on Image Processing, ICIP-95, vol. 1, 1995, pp. 53–56.

[12] K. Engan, Frame based signal representation and compression, Ph.D. Thesis, Norges teknisk-naturvitenskapelige universitet (NTNU)/Høgskolen i Stavanger, available at http://www.ux.his.no/~kjersti/ September 2000.

[13] A. Berg, W. Mikhael, An efficient structure and algorithm for image representation using nonorthogonal basis images, IEEE Trans. Circuits Syst. II 44 (10) (1997) 818–828.

[14] M. Gharavi-Alkhansari, A model for entropy coding in matching pursuit, in: Proceedings of the IEEE International Conference on Image Processing, ICIP-98, vol. 1, 1998, pp. 778–782.

[15] H. Li, I. Wollf, Multiscale matching pursuit for image coding, in: Proceedings of the ISSPA 1999, Brisbane, Australia, 1999, pp. 805–808.

[16] M. Gharavi-Alkhansari, A fast globally optimal algorithm for template matching using low-resolution pruning, IEEE Trans. Image Process. 10 (4) (2001) 526–533.

[17] C.-L. Huang, S.-H. Hsu, Road sign interpretation using matching pursuit method, in: Proceedings of the 4th IEEE Southwest Symposium on Image Analysis and Interpretation, 2000, pp. 202–206.

[18] P.J. Durka, P.J. Ircha, K.J. Blinowska, Stochastic time-frequency dictionaries for matching pursuit, IEEE Trans. Signal Process. 49 (3) (2001) 507–510.

[19] R. Gribonval, Fast matching pursuit with a multiscale dictionary of Gaussian chirps, IEEE Trans. Signal Process. 49 (5) (2001) 994–1001.

[20] Y.-T. Chou, W.-L. Hwang, C.-L. Huang, Matching pursuit low-bit rate video coding with dictionary optimized by shape-gain vector quantizer, available at http://citeseer.nj.nec.com/394375.html, 1999.

[21] B.A. Olshausen, D.J. Field, Sparse coding with an overcomplete basis set: a strategy employed in V1, Vision Research 37 (1997) 3311–3325.

[22] M.S. Lewicki, T.J. Sejnowski, Learning overcomplete representations, Neural Computation 12 (2000) 337–365.

[23] R.J. Duffin, R.C. Schaeffer, A class of nonharmonic Fourier series, Transactions of the American Mathematical Society 72 (1952) 341–366.

[24] C.E. Heil, D.F. Walnut, Continuous and discrete wavelet transforms, SIAM Rev. 31 (4) (1989) 628–666.

[25] I. Daubechies, The wavelet transform, time-frequency localization and signal analysis, IEEE Trans. Inform. Theory 36 (5) (1990) 961–1005.

[26] T.A. Ramstad, S.O. Aase, J.H. Husøy, Subband Compression of Images—Principles and Examples, Elsevier Science Publishers BV, North Holland, 1995.

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

[28] B.K. Natarajan, Sparse approximate solutions to linear systems, SIAM J. Comput. 24 (1995) 227–234.

[29] S.F. Cotter, J. Adler, B.D. Rao, K. Kreutz-Delgado, Forward sequential algorithms for best basis selection, IEE Proc. Vis. Image Signal Process 146 (5) (1999) 235–244.

[30] M. Gharavi-Alkhansari, T.S. Huang, A fast orthogonal matching pursuit algorithm, in: Proceedings of the ICASSP '98, Seattle, USA, 1998, pp. 1389–1392.

[31] L.N. Trefethen, D. Bau, Numerical Linear Algebra, SIAM, Philadelphia, PA, USA, 1997.

[32] P.P. Vaidyanathan, Multirate Systems and Filter Banks, Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.

[33] Z. Cvetković, M. Vetterli, Oversampled filter banks, IEEE Trans. Signal Process. 46 (5) (1998) 1245–1255.

[34] H. Bölcskei, F. Hlawatsch, H.G. Feichtinger, Frame-theoretic analysis of oversampled filter banks, IEEE Trans. Signal Process. 46 (12) (1998) 3256–3268.

[35] K. Skretting, Sparse signal representation using overlapping frames, Ph.D. Thesis, NTNU Trondheim and Høgskolen i Stavanger, available at http://www.ux.his.no/~karlsk/ October 2002.

[36] M.I. of Technology, The MIT-BIH Arrhythmia Database CD-ROM, second ed., MIT, 1992.

[37] H.S. Malvar, D.H. Staelin, The LOT: transform coding without blocking effects, IEEE Trans. Acoust. Speech, Signal Process. 37 (4) (1989) 553–559.