Fast Matching Pursuit with Multi-Gabor Dictionaries

Zdeněk Průša

***Abstract*—Finding the best K-sparse approximation of a signal in a redundant dictionary is an NP-hard problem significant for many applications. Suboptimal greedy matching pursuit (MP) algorithms are generally used for such task. In this work, we present an acceleration technique of the matching pursuit algorithm acting on a multi-Gabor dictionary; a concatenation of several Gabor-type time-frequency dictionaries; each of which consisting of translations and modulations of a possibly different prototype window and time and frequency shift parameters. The technique is based on pre-computing and thresholding inner products between atoms and on updating the residuum and the approximation error estimate directly in the coefficient domain i.e. without the round-trip to the signal domain. We provide an implementation written in C (compatible with C99 and C++11) and we also provide Matlab and GNU Octave wrappers. For some settings, the implementation is almost 100 times faster than the standard Matching Pursuit Toolkit (MPTK).**

I. Introduction

The best K-sparse approximation of a signal $\mathbf{x} \in \mathbb{R}^L$ in an overcomplete dictionary of P normalized atoms (vectors) $\mathbf{D} = [\mathbf{d}_0|\mathbf{d}_1|\dots|\mathbf{d}_{P-1}] \in \mathbb{C}^{L\times P}$, $\|\mathbf{d}_p\|_2 = 1$ is an NP-hard problem [1]. Given the budget of K nonzero elements of the coefficient vector $\mathbf{c} \in \mathbb{C}^P$, the problem can be formally written as the minimization of the approximation error in the energy norm $\|\mathbf{x} - \mathbf{D}\mathbf{c}\|_2$ such that

$$\min\|\mathbf{x} - \mathbf{D}\mathbf{c}\|_2 \quad \text{subject to} \quad \|\mathbf{c}\|_0 \leq K, \tag{1}$$

where the "zero" norm $\|\cdot\|_0$ returns the number of non-zero elements. A similar problem is the minimization of $\|\mathbf{c}\|_0$ given the approximation error tolernace E

$$\min\|\mathbf{c}\|_0 \quad \text{subject to} \quad \|\mathbf{x} - \mathbf{D}\mathbf{c}\|_2 \leq E. \tag{2}$$

Both problems can be tackled by employing one of greedy *matching pursuit* (MP) algorithms. The only difference is the choice of the stopping criterion. However, greedy algorithms are known to be suboptimal in the sense that they are not guaranteed to choose the best combination of K atoms. Instead, an approximation rate i.e. the decrease of the approximation error with iterations has been studied. It has been shown that the basic version of MP [2] achieves an exponential approximation rate [1], [3], [4], [5]. To date, several variants of the generic MP and its orthogonal version OMP [6], [7] were proposed e.g. complementary MP [8], [9], cyclic MP [10], [11], gradient pursuit [12], [13], local OMP [14], [15] and self projected MP [16]. In practice, without imposing any structure on the dictionary, the effectiveness of the algorithms quickly deteriorates when increasing the dimensionality of the problem; either by increasing the input signal length L or the size of the dictionary P. Even with structured dictionaries, which allow usage of fast algorithms in place of matrix operations, a naive implementation can still be prohibitively inefficient; e.g. processing even just a few seconds of an audio signal, which typically consist of tens of thousands of samples per second, can take hours.

An overview of the greedy algorithms, a class of algorithms MP falls under, can be found in [17], [18] and in the context of audio and music processing in [19], [20], [21]. Notable applications of MP algorithms include audio analysis [22], [23], coding [24], [25], [26], time scaling/pitch shifting [27] [28], source separation [29], denoising [30], partial and harmonic detection and tracking [31] and EEG analysis [32].

In this contribution, we present a method for accelerating MP-based algorithms acting on a single overcomplete Gabor dictionary or on a concatenation of several Gabor dictionaries with possibly different windows and time and frequency steps termed a multi-Gabor dictionary by Li [33] and Wolfe et al. [34]. Note that such systems are more general than multiscale Gabor dictionaries [2] as well as multi-window Gabor dictionaries [35]. Multiscale Gabor dictionary involves only a single window shape and its dilations, while a multi-window Gabor dictionary considers equal time and frequency steps for all windows. The main idea of the present acceleration technique is to exploit the locality of the inner products between the atoms in the dictionaries and dismiss values below user definable threshold. It is then feasible to store all significant inner products in a lookup table and avoid atom synthesis and the residual re-analysis in every iteration of MP as it is usually done in practice. The size of the lookup table is independent of the signal length and it only depends on the parameters of the Gabor dictionaries.

The C code (compatible with C99 and C++11) can be found in the backend library of the Matlab/GNU Octave Large Time-Frequency Analysis Toolbox (LTFAT, http://ltfat.github.io) [36], [37] available individually at http://ltfat.github.io/libltfat. The low level C language (or rather a subset of C99 and C++11 standards) was chosen due to two reasons. First, the nature of the MP algorithm does not lend itself to an efficient implementation in a high level language due to its overhead. For example, a proof-of-concept Matlab implementation was about 50 times slower than the final C implementation. Second, a C-based shared (dynamic) library can be interfaced from most of the high and even low level languages. The programming in-

Z. Průša* is with the Acoustics Research Institute, Austrian Academy of Sciences, Wohllebengasse 12–14, 1040 Vienna, Austria, email: zdenek.prusa@oeaw.ac.at (corresponding address).

This work was supported by the Austrian Science Fund (FWF): Y 551–N13 and I 3067–N30.

terface (documentation available at http://ltfat.github.io/libltfat/group__multidgtrealmp.html) was designed with this use case in mind. Indeed, since version 2.3.0, LTFAT itself interfaces the library trough a MEX function, whose call is wrapped in a function `multidgtrealmp` (see http://ltfat.github.io/doc/gabor/multidgtrealmp.html).

A. Relationship to Previous Work

To date, considering a wast body of literature dealing with MP, surprisingly few authors address effective (non-textbook) implementation of the algorithm let alone provide code. In the original paper, Mallat and Zhang [2, Appendix E] proposed to perform the residual update in the coefficient domain using inner products between the atoms. They present an analytic formula for evaluating the inner products between atoms of a multi-scale Gabor dictionary with a Gaussian window. An implementation by Ferrando et. al. [38] is tailored to the Gaussian window-based multiscale Gabor dictionary defined on an interval. The authors choose to trade updating the residuum directly in the coefficient domain for the flexibility in choosing the dictionary parameters and in boundary handling. The defacto standard implementation of several MP based algorithms is in the Matching Pursuit Toolkit (MPTK) [39]. The toolbox is not restricted to Gabor dictionaries, and, therefore, the "update in the coefficient domain"-trick is not exploited.

In comparison, the present method and implementation is applicable to general multi-Gabor dictionaries while being much faster than MPTK.

B. Notation

Matrices will be denoted with bold capital upright letters, e.g., $\mathbf{M}$, column vectors with lowercase bold upright letter such as $\mathbf{x}$. Conjugate transpose will be denoted with a star superscript, $(\mathbf{x}^*, \mathbf{M}^*)$, scalar variables with a capital or lowercase italics letter s, S and scalar constants as upright capital or lowercase letters like $\pi, \mathrm{e}, \mathrm{i}$. A single element of a matrix or a vector will be selected using round brackets $\mathbf{M}(m,n)$, $\mathbf{x}(l)$. The index is always assumed to be applied modulo vector length (or matrix size in the respective direction) such that $\mathbf{x}(l) = \mathbf{x}(l + kL)$ for $l = 0, \ldots, L-1$ and $k \in \mathbb{Z}$. Moreover, we will use two indices for vectors such that $\mathbf{c}(m,n)_M = \mathbf{c}(m+nM)$ in order to transparently "matrixify" a vector. Sub-vectors and sub-matrices will be selected by an index set denoted by a caligraphic letter e.g. $\mathbf{x}(\mathcal{P})$ and entire rows or columns of a matrix will be selected using "the Matlab notation" e.g. the m-th row of a matrix $\mathbf{M}$ is $\mathbf{M}(m,:)$ and n-th column is $\mathbf{M}(:,n)$. We will omit brackets when indexing the outcome of the matrix-vector or matrix-matrix product i.e. we will use $\mathbf{M}\mathbf{x}(p)$ instead of $(\mathbf{M}\mathbf{x})(p)$. Scalar-domain functions used on matrices or vectors are applied element-wise e.g. $|\mathbf{x}|^2(l) = |\mathbf{x}(l)|^2$. The inner product of two vectors in $\mathbb{C}^L$ is given as $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{y}^*\mathbf{x} = \sum_{l=0}^{L-1} \mathbf{x}(l)\overline{\mathbf{y}(l)}$, where the overline denotes complex conjugation. Real and imaginary parts of a complex number will be denoted as $\mathrm{Re}(c)$ and $\mathrm{Im}(c)$ respectively, and the phase as $\arg(c)$ such that $c = \mathrm{Re}(c) + \mathrm{i}\,\mathrm{Im}(c) = |c|\mathrm{e}^{\mathrm{i}\arg(c)}$. The k–norm of a vector is defined as $\|\mathbf{x}\|_k = (\sum_{l=0}^{L-1} |\mathbf{x}(l)|^k)^{1/k}$. In particular, the 2–norm relates to the inner product as $\|\mathbf{x}\|_2^2 = \langle \mathbf{x}, \mathbf{x} \rangle$. The best approximation given specific K atoms with the index set $\mathcal{P}$ denoting the submatrix as $\mathbf{D}_\mathcal{P} = \mathbf{D}(:,\mathcal{P})$ can be obtained by the orthogonal projection

$$\mathbf{x}_\mathcal{P} = \mathbf{D}_\mathcal{P}\mathbf{D}_\mathcal{P}^+\mathbf{x} = \mathbf{D}_\mathcal{P}\left(\mathbf{D}_\mathcal{P}^*\mathbf{D}_\mathcal{P}\right)^+\mathbf{D}_\mathcal{P}^*\mathbf{x}, \tag{3}$$

where $^+$ denotes the Moore-Penrose pseudoinverse. The formula simplifies to

$$\mathbf{x}_\mathcal{P} = \mathbf{D}_\mathcal{P}\left(\mathbf{D}_\mathcal{P}^*\mathbf{D}_\mathcal{P}\right)^{-1}\mathbf{D}_\mathcal{P}^*\mathbf{x} \tag{4}$$

if the atoms are linearly independent.

Circular translation and modulation will be denoted as matrix operations $\mathbf{T}_n\mathbf{x}(l) = \mathbf{x}(l-n)$ and $\mathbf{M}_m\mathbf{x}(l) = \mathbf{x}(l)\mathrm{e}^{\mathrm{i}2\pi ml}$ respectively.

C. Multi-Gabor Dictionaries

A Gabor dictionary $\mathbf{D}_{(\mathbf{g},a,M)}$ generated from a window $\mathbf{g} \in \mathbb{R}^L, \|\mathbf{g}\|_2 = 1$, time shift a and a number of modulations M is given as

$$\begin{aligned}\mathbf{D}_{(\mathbf{g},a,M)}(l, m+nM) = \mathbf{d}_{m+nM}(l) &= \mathbf{T}_{na}\mathbf{M}_{m/M}\mathbf{g}(l) \\ &= \mathbf{g}(l-na)\mathrm{e}^{\mathrm{i}2\pi m(l-na)/M}\end{aligned} \tag{5}$$

for $l = 0, \ldots, L-1$ and $m = 0, \ldots, M-1$ for each $n = 0, \ldots, N-1$, where $N = L/a$ is the number of window time shifts and the overall number of atoms is $P = MN$. The expression $(l - na)$ is assumed to be evaluated modulo L according to the circular indexing. The length of the window will further be denoted by $\mathrm{len}(\mathbf{g}) \leq L$ and the redundancy of a dictionary will be defined as $P/L = M/a$. A multi-Gabor dictionary consisting of W Gabor dictionaries is defined as

$$\left[\mathbf{D}_{(\mathbf{g}_1,a_1,M_1)}\middle|\mathbf{D}_{(\mathbf{g}_2,a_2,M_2)}\middle|\ldots\mathbf{D}_{(\mathbf{g}_W,a_W,M_W)}\right]. \tag{6}$$

Due to technical reasons explained in Sec. II-B, we restrict parameters a_w and M_w such that every pair a_u, a_v and M_u, M_v is divivible by $a_{\min} = \min\{a_u, a_v\}$ and $M_{\min} = \min\{M_u, M_v\}$ respectivelly. Further, we assume the parameters are chosen such that each subdirectory is a frame [40] for $\mathbb{C}^L$. That means each subdirectory $\mathbf{D}_w$ has full row rank, the atoms span the whole $\mathbb{C}^L$ and each $\mathbf{D}_w$ is right-invertible. Although such strong assumption is not strictly necessary, it is amost always satisfied for commonly used Gabor dictionaries and even allows dictionaries with windows being longer than the number of frequency channels i.e. $\mathrm{len}(\mathbf{g}) > M$. A simple check for whether a Gabor dictionary specified by $\mathbf{g}, a, M$ ($M/a = k \in \mathbb{Z}^+$) satisfies frame conditions is based on the Walnut factorization of the window [41], [42]. For the purpose of the frame check, assume the window $\mathbf{g}$ is zero-padded to length L_M, the next integer multiple of M (such that $L_M = QM$), and it is then circularly shifted such that the "peak" resides at index 0. Then the following expression

$$g(l,q) = \sum_{n=0}^{M/a-1}\left|\sum_{r=0}^{Q-1}\mathbf{g}\left(l - na + rM\right)\mathrm{e}^{-\mathrm{i}2\pi rq/Q}\right|^2, \tag{7}$$

must obey $0 < A \leq g(l,q) \leq B$ for $l = 0, \ldots, a-1$ and $q = 0, \ldots, Q-1$. The index of $\mathbf{g}$ is assumed to be evaluated

modulo L_M. Note that in the special "painless" case $\text{len}(\mathbf{g}) \leq M$ ($Q = 1$), the formula becomes

$$g(l) = \sum_{n=0}^{M/a-1} |\mathbf{g}\,(l - na)|^2 . \tag{8}$$

This is equivalent to a more common test involving a sum of all possible squared window translations that must be bounded between A, B such that $0 < A \leq \left(\sum_n \mathbf{T}_{na}|\mathbf{g}|^2\right)(l) \leq B$ for all $l = 0, \ldots, L-1$. The ratio B/A is actually equal to the squared condition number of $\mathbf{D}_{(\mathbf{g},a,M)}$ and a high condition number signalizes a "bad" frame i.e. numerical precision issues should be expected.

D. Matching Pursuit – MP

In this section we recall the idea behind the MP algorithm, summarize its steps and explain an alternative way of performing the MP iterations exploiting the inner products between the atoms.

The MP algorithm iteratively decreases the approximation error (energy of the residuum) $E_{k+1} = \|\mathbf{r}_{k+1}\|_2^2$ by considering orthogonal projections of the residuum $\mathbf{r}_k$ over the individual P elements of the normalized dictionary $\langle \mathbf{r}_k, \mathbf{d}_p\rangle \mathbf{d}_p$, selecting the one ($p_{\text{max}}$) which decreases the energy of the residual $\|\mathbf{r}_k - \langle \mathbf{r}_k, \mathbf{d}_p\rangle \mathbf{d}_p\|_2^2$ the most and updating the residuum $\mathbf{r}_{k+1} = \mathbf{r}_k - \langle \mathbf{r}_k, \mathbf{d}_{p_{\text{max}}}\rangle \mathbf{d}_{p_{\text{max}}}$. Since the energy of the new potential residual can be written as

$$\|\mathbf{r}_k - \langle \mathbf{r}_k, \mathbf{d}_p\rangle \mathbf{d}_p\|_2^2 = \|\mathbf{r}_k\|_2^2 - |\langle \mathbf{r}_k, \mathbf{d}_p\rangle|^2, \tag{9}$$

the best atom to choose is the one with the highest inner product with the residuum i.e.

$$p_{\text{max}} = \underset{p}{\text{argmax}}\, |\langle \mathbf{r}_k, \mathbf{d}_p\rangle|.$$

The procedure is repeated until desired approximation error is achieved or alternatively some other stopping criterion is met e.g. a sparsity limit it reached. Formally, the algorithm is summarized in Alg. 1.

Input: Input signal $\mathbf{x}$, dictionary $\mathbf{D}$
Output: Solution vector $\mathbf{c}$
Initialization: $\mathbf{c} = \mathbf{0}$, $\mathbf{r}_0 = \mathbf{x}$, $E_0 = \|\mathbf{r}_0\|_2^2$, $k = 0$
while *Stopping criterion not met* **do**
1) **Selection:**
$p_{\text{max}} = \underset{p}{\text{argmax}}\, |\langle \mathbf{r}_k, \mathbf{d}_p\rangle|, \quad c_{\text{max}} = \langle \mathbf{r}_k, \mathbf{d}_{p_{\text{max}}}\rangle$
2) **Update:**
a) **Solution:** $\mathbf{c}(p_{\text{max}}) = \mathbf{c}(p_{\text{max}}) + c_{\text{max}}$
b) **Error:** $E_{k+1} \leftarrow E_k - |c_{\text{max}}|^2$
c) **Residual:** $\mathbf{r}_{k+1} = \mathbf{r}_k - c_{\text{max}}\mathbf{d}_{p_{\text{max}}}$
$k \leftarrow k + 1$
end

Algorithm 1: Basic Matching Pursuit

The error is usually normalized and converted to decibels by $10 \log_{10} E_{k+1}/\|\mathbf{x}\|_2^2$. It is known that the matching pursuit (MP) algorithm and its derivatives can benefit from precomputing inner products between the atoms in the dictionary $\mathbf{G}(i,j) = \langle \mathbf{d}_j, \mathbf{d}_i\rangle$ i.e. from pre-computing the Gram matrix $\mathbf{G} = \mathbf{D}^*\mathbf{D} \in \mathbb{C}^{P\times P}$. The residual update step 2c is then written as [43, Ch. 12]

$$\mathbf{D}^*\mathbf{r}_{k+1} = \mathbf{D}^*\mathbf{r}_k - c_{\text{max}}\mathbf{G}(:, p_{\text{max}}). \tag{10}$$

This modification has the advantage of removing the necessity of synthesizing the residual and of recomputing of the inner product in the selection step. On the other hand, such approach is usually dismissed as impractical in the literature due to the high memory requirements for storing the Gram matrix. This is however not the case for a well behaved multi-Gabor dictionary, for which the Gram matrix can be precomputed and significant values stored compactly.

E. Real Atoms From A Complex Dictionary

The MP algorithm with a complex Gabor dictionary as discussed so far is valid for complex as well as for real-valued signals. However, when dealing with real signals, one might wish to work with real atoms exclusively such that the signal approximation and the residual are real at any stage of the algorithm and the positive-negative frequency conjugate symmetry of the coefficients is preserved. In effect "half" of the dictionary could be disregarded. To achieve that, pairs of conjugated atoms must be considered such that instead of performing projections onto individual atoms the selection step involves orthogonal projections of the residual onto subspaces spanned by conjugated pairs of atoms [19]. For each pair of conjugated atoms arranged into a matrix $\mathbf{P} = [\mathbf{d}|\overline{\mathbf{d}}] \in \mathbb{C}^{L\times 2}$, the projection can be written as $\mathbf{P}(\mathbf{P}^*\mathbf{P})^{-1}\mathbf{P}^*\mathbf{r}_k$ (according to (4)) and the energy of a potential new residual becomes $\|\mathbf{r}_k - \mathbf{P}(\mathbf{P}^*\mathbf{P})^{-1}\mathbf{P}^*\mathbf{r}_k\|_2^2$. Due to the orthogonality of the projection, the pair which decreases the residual error the most is the one with highest $\|\mathbf{P}(\mathbf{P}^*\mathbf{P})^{-1}\mathbf{P}^*\mathbf{r}_k\|_2$. The inverted expression can be written as

$$\begin{aligned}(\mathbf{P}^*\mathbf{P})^{-1} &= \begin{bmatrix} 1 & \langle \overline{\mathbf{d}}, \mathbf{d}\rangle \\ \langle \mathbf{d}, \overline{\mathbf{d}}\rangle & 1 \end{bmatrix}^{-1} \\ &= \frac{1}{1 - |\langle \mathbf{d}, \overline{\mathbf{d}}\rangle|^2} \begin{bmatrix} 1 & -\langle \mathbf{d}, \overline{\mathbf{d}}\rangle \\ -\overline{\langle \mathbf{d}, \overline{\mathbf{d}}\rangle} & 1 \end{bmatrix}\end{aligned} \tag{11}$$

and the *dual* pair of conjugated atoms is obtained as

$$\begin{aligned}\widetilde{\mathbf{P}} &= [\widetilde{\mathbf{d}} \mid \overline{\widetilde{\mathbf{d}}}] = \mathbf{P}(\mathbf{P}^*\mathbf{P})^{-1} \\ &= \frac{1}{1 - |\langle \mathbf{d}, \overline{\mathbf{d}}\rangle|^2} \left[\mathbf{d} - \overline{\langle \mathbf{d}, \overline{\mathbf{d}}\rangle}\overline{\mathbf{d}} \;\middle|\; \overline{\mathbf{d}} - \langle \mathbf{d}, \overline{\mathbf{d}}\rangle \mathbf{d}\right].\end{aligned} \tag{12}$$

The energy of the projection can be written as

$$\begin{aligned}\|\mathbf{P}(\mathbf{P}^*\mathbf{P})^{-1}\mathbf{P}^*\mathbf{r}_k\|_2^2 &= \|\langle \mathbf{r}_k, \widetilde{\mathbf{d}}\rangle \mathbf{d} + \langle \mathbf{r}_k, \overline{\widetilde{\mathbf{d}}}\rangle \overline{\mathbf{d}}\|_2^2 \\ &= 2|\langle \mathbf{r}_k, \widetilde{\mathbf{d}}\rangle|^2 \left(1 + \text{Re}(\mathrm{e}^{\mathrm{i}2\phi}\langle \mathbf{d}, \overline{\mathbf{d}}\rangle)\right),\end{aligned} \tag{13}$$

where $\phi = \arg(\langle \mathbf{r}_k, \widetilde{\mathbf{d}}\rangle)$. The inner product of the residual with the first of the dual atoms $\widetilde{\mathbf{d}}$ can be expanded into

$$\widetilde{c} = \langle \mathbf{r}_k, \widetilde{\mathbf{d}}\rangle = \frac{\langle \mathbf{r}_k, \mathbf{d}\rangle - \langle \mathbf{d}, \overline{\mathbf{d}}\rangle \overline{\langle \mathbf{r}_k, \mathbf{d}\rangle}}{1 - |\langle \mathbf{d}, \overline{\mathbf{d}}\rangle|^2}, \tag{14}$$

and the inner product with the conjugated one is just conjugated. Equation (13) can be rewritten without the costly exp and arg operations using only real products and additions as

$$\begin{aligned}&\|\mathbf{P}(\mathbf{P}^*\mathbf{P})^{-1}\mathbf{P}^*\mathbf{r}_k\|_2^2 = \\ &2\left[\mathrm{Re}(\widetilde{c})^2+\mathrm{Im}(\widetilde{c})^2+\mathrm{Re}(\langle\mathbf{d},\overline{\mathbf{d}}\rangle)\left(\mathrm{Re}(\widetilde{c})^2-\mathrm{Im}(\widetilde{c})^2\right)-\right. \\ &\left.-\,2\,\mathrm{Im}(\langle\mathbf{d},\overline{\mathbf{d}}\rangle)\,\mathrm{Re}(\widetilde{c})\,\mathrm{Im}(\widetilde{c})\right].\end{aligned} \quad (15)$$

Obviously, as long as $\langle\mathbf{d},\overline{\mathbf{d}}\rangle \approx 0$, we can simplify the equations such that it is enough to consider only a single atom $\mathbf{d}$. A special case are atoms with frequency index $m = 0$, as well as $m = M/2$ if M is even, which are real already and do not have a conjugated parther.

II. Faster Approximate Coefficient-Domain Residual Update

As already mentioned, the acceleration technique works with the coefficient domain update formula (10). In this section, we will show that significant values of the Gram matrix $\mathbf{G}$ can be precomputed using possibly much shorter signal length than given L, truncated and stored efficiently for a single as well as for a multi-Gabor dictionary. We exploit the fact that the Gram matrix of a single Gabor dictionary is highly structured. In fact, it takes the form of a *twisted convolution* [44] matrix with a fixed kernel $\mathbf{h} = \mathbf{D}^*\mathbf{g} \in \mathbb{C}^{MN}$. Therefore the columns of the Gram matrix are constructed by shifting and modulating the kernel such that

$$\mathbf{G}(m+nM,k+jM) = \mathbf{h}(m-k,n-j)_M \mathrm{e}^{\mathrm{i}2\pi k\frac{a}{M}(n-j)} \quad (16)$$

for $m = 0,\dots,M-1$ for each $n = 0,\dots,N-1$ and $k = 0,\dots,M-1$ for each $j = 0,\dots,N-1$. With respect to indexing $\mathbf{h}(m,n)_M$, the kernel is essentially supported around the origin for well-behaved low-pass windows and, as it turns out, it can be truncated and stored efficiently. Therefore, the residual update (10) reduces to a subtraction of a truncated, modulated and weighted kernel

$$\mathbf{h}^{(k,j)}(m,n)_M = \mathbf{h}(m,n)_M \mathrm{e}^{\mathrm{i}2\pi k\frac{a}{M}n} \quad (17)$$

from the neighborhood of the time-frequency position k,j ($p_{\max} = k + jM$) which can be written as

$$\mathbf{D}^*\mathbf{r}_{k+1}(\mathcal{M},\mathcal{N})_M = \mathbf{D}^*\mathbf{r}_k(\mathcal{M},\mathcal{N})_M - c_{\max}\mathbf{h}^{(k,j)} \quad (18)$$

assuming $\mathbf{h}$ has already been truncated and $\mathcal{M}$ and $\mathcal{N}$ denote the index set of the appropriate neighborhood.

A. Pre-computing Kernel for Single Dictionary

The inner products between the window and its translations which are sufficiently far away are obviously zero or at least negligible. Therefore after determining the length of the window effective support, one can compute the minimum admissible $L_{\min}$ for the computation of values of kernel $\mathbf{h}$ as being the next integer multiple of M bigger than twice the length of the support. The kernel is further truncated also in the frequency direction. Examples of abs. values of kernels for several windows using time shift $a = 512$ and $M = 2048$ frequency channels are depicted in Fig. 1. The values are in dB relative to the maximum element with 0 dB. Values below

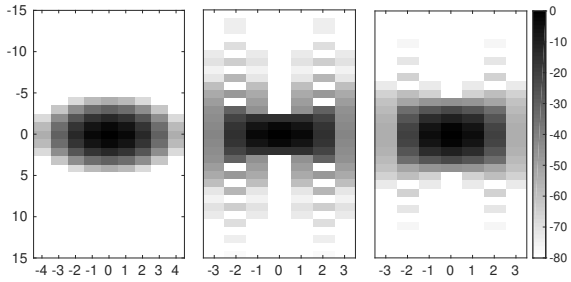

Figure 1: Examples of abs. values of truncated kernels. (Left) Gaussian, (middle) Hann and (right) Blackman windows.

-80 dB were cropped. The idea of truncating the kernel $\mathbf{h}$ comes from Le Roux et al. [45] who used it for replacing the operation of the (cross) Gram matrix in an iterative scheme dealing with magnitude-only reconstruction. The authors of the aforementioned paper also noticed that the kernel is conjugate symmetric about both the horizontal (time) and the vertical (frequency) axes, which could be exploited for reducing the number of multiplications further. When inspecting formula (17), it is obvious that for a fixed time-frequency position k,j the modulation by $2\pi ka/M$ radians is performed on all rows of the kernel independently. Moreover, the modulation frequencies are M/a periodic in k and, therefore, all M/a unique complex exponentials can be tabulated and stored. The memory requirements are equal to storing M/a additional rows of the kernel. The cost of applying the modulation during the residual update step is one complex multiplication per kernel column.

B. Pre-computing Cross-Kernels Between Dictionaries

The Gram matrix of a multi-Gabor dictionary consists of Gram matrices of individual dictionaries $\mathbf{D}_w$ and cross-Gram matrices [40] between the dictionaries. Denoting a cross-Gram matrix as $\mathbf{G}_{w,v} = \mathbf{D}_w^*\mathbf{D}_v$ the overall Gram matrix is a block matrix with the following structure

$$\begin{bmatrix}\mathbf{G}_{1,1} & \mathbf{G}_{1,2} & \dots & \mathbf{G}_{1,W}\\ \mathbf{G}_{2,1} & \mathbf{G}_{2,2} & \dots & \mathbf{G}_{2,W}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{G}_{W,1} & \mathbf{G}_{W,2} & \dots & \mathbf{G}_{W,W}\end{bmatrix}. \quad (19)$$

A cross-Gram matrix $\mathbf{G}_{w,v}$ shares the same twisted convolution structure with the regular Gram matrix with kernel $\mathbf{h}_{w,v} = \mathbf{D}_w^*\mathbf{g}_v$ only if the time-frequency shifts are equal i.e. $a_w = a_v$ and $M_w = M_v$ (multi-window Gabor dictionary case). In the case the parameters differ and we are therefore dealing with a multi-Gabor dictionary, the twisted convolution structure is lost. Nevertheless, as long as all four parameters a_w, a_v, M_w and M_v are integer multiples of $a_{\min} = \min\{a_w, a_v\}$, the cross-Gram matrix can still be represented efficiently by a kernel computed for a "common" time-frequency grid defined by parameters $a_{\min}$ and $M_{\max} = \max\{M_w, M_v\}$. In the residual update step of the inner products of the residual with the w-th dictionary, the modulated kernel is subsampled by ratios $a_w/a_{\min}$ and $M_{\max}/M_w$ in horizontal and vertical directions respectively. To illustrate, consider a multi-Gabor dictionary

consisting of two Gabor dictionaries $\mathbf{D}_1 = \mathbf{D}_{(\mathbf{g}_1,a_1,M_1)}$ and $\mathbf{D}_2 = \mathbf{D}_{(\mathbf{g}_2,a_2,M_2)}$ with $a_1 = 4a_{\text{min}}, M_1 = 8a_{\text{min}}$ and $a_2 = a_{\text{min}}, M_2 = 2a_{\text{min}}$. Both cross-kernels $\mathbf{h}_{1,2}$ and $\mathbf{h}_{2,1}$ are computed with $a_{\text{min}} = a_2$ and $M_{\text{max}} = M_1$. The example in Fig. 2 depicts an update of inner products of the residual with both dictionaries. The upper figure shows the case when a coefficient from the dictionary 1 was selected in the selection step of MP and inner products with dictionary 2 are being updated and vice versa below. The dashed line square depicts the area covered by a cross-kernel with respect to the common grid $a_{\text{min}}, M_{\text{max}}$.

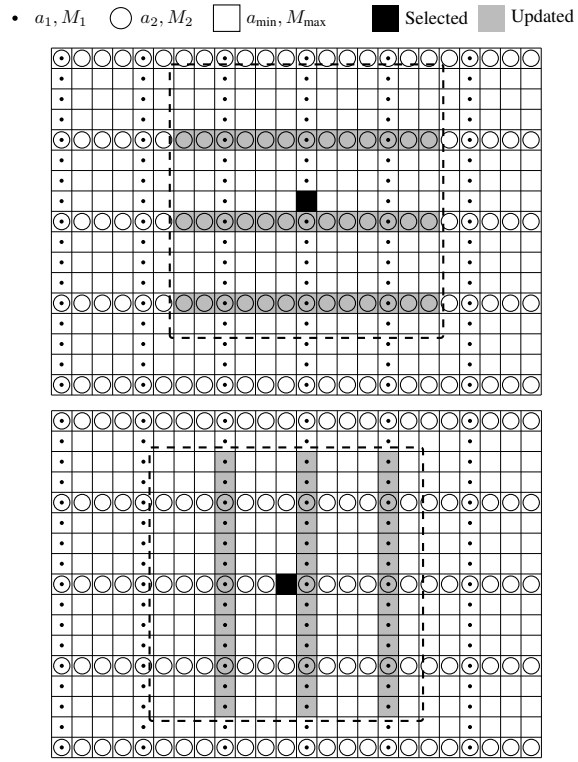

Figure 2: Illustration of the residual update between dictionaries using cross kernels.

III. Practical Consideration

Since MP is a simple algorithm, the main focus will be on a detailed description of the accelerated residual update step from Sec. II. Other steps will be mentioned in less detail. The presented description is valid for the real setting from Sec. I-E, therefore, only the first $M_{w,\mathcal{R}} = \lfloor M_w/2 \rfloor + 1$ frequency channels form each Gabor dictionary w will be considered and the reduced dictionary will be denoted as $\mathbf{D}_{w,\mathcal{R}}$. A real signal can be recovered from reduced coefficient vectors $\mathbf{c}_{w,\mathcal{R}}$ as

$$\mathbf{x} = \sum_{w=1}^{W} \mathbf{D}_{w,\mathcal{R}}\mathbf{c}_{w,\mathcal{R}} + \overline{\mathbf{D}_{w,\mathcal{R}}}\overline{\mathbf{c}_{w,\mathcal{R}}} \tag{20}$$

where elements of $\overline{\mathbf{c}_{w,\mathcal{R}}}$ are set to zero whenever the conjugated partner is missing i.e. for frequency index $m = 0$ and $m = M_w/2$ if M_w is even. Obviously, in practice the matrix operations are replaced by an efficient FFT-based algorithm (see e.g. [46], [42]). In the remainder of this section, we first address the issue of an efficient search for the maximum inner product. In Sec. I-E it is stated that projections onto pairs of conjugated atoms should be taken into account i.e. (15) should be used in comparisons. In practice, discarding the effect of the nonzero inner product between the conjugated atoms in the selection step (setting $\langle \mathbf{d}, \overline{\mathbf{d}} \rangle = 0$) does not have a significant impact on the result and leads to about 20% overall speedup. Our implementation supports both options and the technically correct one will be referred to as *pedantic*. It is however crucial the selected coefficient is adjusted using (14) prior to the residual update step. Finally, due to the conjugate symmetry, the residual update step now involves a pair of atoms each of which can however be treated separately.

In the description of the implementation, we will work exclusively with the reduced dictionary, therefore we will drop the $\mathcal{R}$ subscript in order to lighten the notation.

Given W sets of Gabor dictionary parameters $(\mathbf{g}_w, a_w, M_w)$ for $w = 1, \ldots, W$, the initialization involves pre-computing (or loading) cross-kernels and the complex exponentials according to the rules described in Sec. II. The algorithm itself then starts by computing inner products of the input signal $\mathbf{r}_0 = \mathbf{x}$ (the initial residual) with atoms from all reduced dictionaries $\mathbf{c}_w^{\mathbf{r}_0} = \mathbf{D}_w^* \mathbf{x}$ for all $w = 1, \ldots, W$.

A. Keeping Track Of The Maximum

Performing the full search for the maximum inner product in each selection step is highly inefficient. The authors of MPTK [39] proposed to store positions of maxima for each window time position and organize them in a partial hierarchical tournament-style tree. Such tree contains at each level maxima from pairs from one level below. Since the residual update affects only a limited number of neighboring time positions, a bottom-up tree update becomes more efficient than a full search. To quantify the reduction of the number of comparisons required to find maximum, consider an array of length L and a tree of depth d, where $d = 0$ means a fallback to a full array search. After the tree has been initialized, the search can be performed at the top level of the tree which requires only $\lceil L/2^d \rceil - 1$ comparisons. When Q consecutive elements from the array are modified, the bottom up tree update requires additional $Q + d$ comparisons in the worst case. Since Q is expected to be much smaller than L, the tree-search is more efficient than a simple search in the whole array which obviously requires $L-1$ comparisons. The worst-case bottom-up update of a 3–level tree is depicted in Fig. 3.

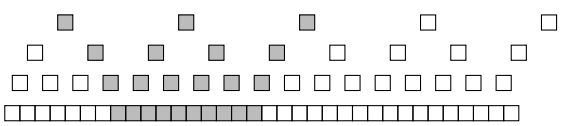

Figure 3: An example of the worst-case 10 element bottom-up tournament tree update.

Moreover, in the case of the present method, the kernel and therefore the residual coefficient update is localized in frequency as well. Therefore, tournament-style trees are used

for keeping track of maxima for individual window time positions (across frequency channels) in the similar manner.

The trees produce the inner product of the residual with atom $\mathbf{d}_{\max}$ denoted as $c_{\max} = \langle \mathbf{r}_k, \mathbf{d}_{\max} \rangle$ from dictionary $w_{\max}$ at position $m_{\max}, n_{\max}$.

B. Fast Update Step With Real Atoms From Complex Multi-Gabor Dictionary

When dealing with real atoms, pairs of conjugated atoms are involved. The inner product of the currently selected atom with the conjugated partner $\langle \mathbf{d}_{\max}, \overline{\mathbf{d}}_{\max} \rangle$ can actually be extracted from the kernel $\mathbf{h}_{w_{\max}, w_{\max}}$. Note that since the kernel has been truncated, we consider the inner product to be zero if the conjugated atom is not in the range of the kernel update. We consider the inner product to be zero also if the conjugated atom is missing.

Before the update step, $c_{\max}$ is adjusted using (14). The solution update step is performed as in Alg. 1 step 2a while the error update step 2b uses (15) in place of the squared magnitude of the coefficient if the conjugated parther of $\mathbf{d}_{\max}$ is present. The steps of the coefficient residual update in the coefficient domain are summarized in Alg. 2. Note the substraction of the kernel is performed for the conjugated atom as well.

IV. Comparison with MPTK

In order to showcase the efficiency of the proposed algorithm and its implementation, in this section, we present a comparison with MPTK (version 0.7.0), which is considered to be the fastest implementation available. We measure the duration of the matching pursuit decomposition only. From MPTK, we used the modified `mpd` utility tool. The testing signal was the first channel from the file no. 39 from the SQAM database [47], which is a 137 seconds long piano recording sampled at 44.1 kHz totaling $6 \cdot 10^6$ samples. Both implementations link the same FFTW library [48] version 3.3 and were compiled using the GCC (g++) compiler (version 7.2.0) with the `-Ofast` optimization flag enabled. The creation of the FFTW plans and the computation of the kernels was excluded from the measurements. In our implementation, the kernels were truncated with relative threshold 10^{-4}, which allows achieving at least -40 dB approximation error in general. The specification of the PC the timing was performed on was Intel® Core™ i5-4570 3.20 GHz, 16 GB RAM running Ubuntu 16.04. The timing was done on an idle machine using the high-precision timer from the C++11 standard library `chrono`. The data type was double precision floating point. We used single and multi Gabor dictionaries with the Blackman window and various redundancies M_w/a_w. The length of the window was always equal to M. In the decomposition we performed $2 \cdot 10^5$ iterations and the execution times in the graphs are in seconds. Figure 4 shows a comparison of execution times for a single dictionary, numbers of channels $M_w = 512, \ldots, 8192$ (additionally also 16384 for the proposed implementation) and various hop sizes a (redundancies). Figure 5 shows a comparison of execution times using two multi-dictionaries each of which consisting of five dictionaries.

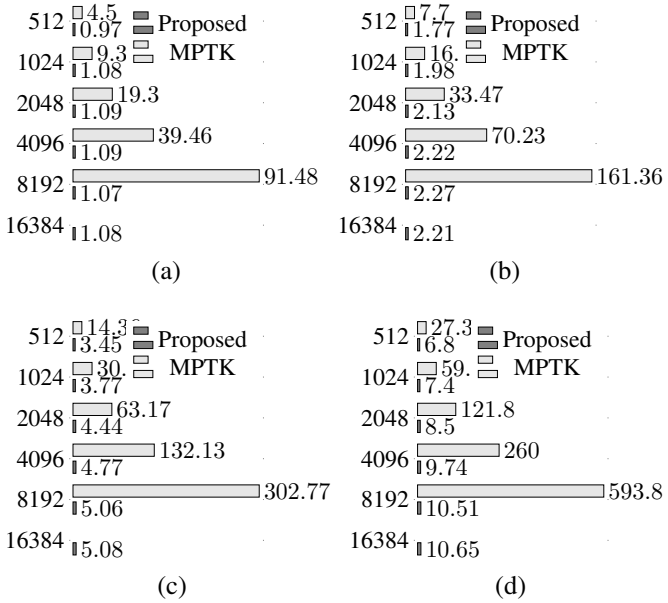

Figure 4: Single Gabor dictionary for different M and various redundancies: (a) $a = M/4$ (b) $a = M/8$ (c) $a = M/16$ (d) $a = M/32$.

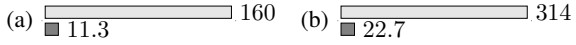

Figure 5: Five Gabor dictionaries with $M_w = 512, \ldots, 8192$, (a) $a_w = M_w/4$, (b) $a_w = M_w/8$

V. Conclusion And Outlook

We have presented a fast algorithm and a reference implementation of an accelerated MP algorithm suitable for multi-Gabor dictionaries. Due to the structure of the Gram matrix of the multi-Gabor dictionary, the coefficient domain residual update step becomes very fast while the memory requirements for storing the inner products between the atoms remain constant with increasing signal length. Moreover, the time and frequency locality of the residual update in turn allows faster search for the maximum in the next iteration. The benchmarks show that, depending on the dictionary, our implementation is 2–90 times faster than the standard MPTK implementation. In the single dictionary case, the most notable feature is that the execution time is virtually independent of the number of channels M when the redundancy M/a is fixed. Moreover, as it turned out, MPTK could not handle dictionaries with the number of channels higher than 8192. In our code, no explicit optimization techniques like exploiting the SIMD operations or parallelization of the code were used, therefore it is possible that there is still room for improvement.

Since the presented acceleration technique applies only to the update step of the algorithm, it is also applicable to various extensions of MP like the molecular MP [49], perceptual MP [50], [26], and guided MP [21] etc. Extensions to the Local OMP [14], [15], cyclic MP [10], [11] and to the complementary MP [8], [9] seem to be possible as well.

In future work, we will investigate the suitability of the

Input: $c_{\max}$, $w_{\max}$, $m_{\max}, n_{\max}$, $\langle \mathbf{d}_{\max}, \overline{\mathbf{d}}_{\max} \rangle$, $a_1, \ldots, a_W$, $M_1, \ldots, M_W$, $\mathbf{h}_{w,v}$ $(w, v = 1, \ldots, W)$
Output: Inner products to be updated $\mathbf{c}_1^{\mathbf{r}_k}, \ldots, \mathbf{c}_W^{\mathbf{r}_k}$
for $w = 1, \ldots, W$ **do**
  $\mathbf{c}_w^{\mathbf{r}_k} \leftarrow$`singleDictionaryResidualUpdate(`$m_{\max}$, w, $\mathbf{c}_w^{\mathbf{r}_k}$`)`
  if $|\langle \mathbf{d}_{\max}, \overline{\mathbf{d}}_{\max} \rangle| > 0$ **then**
    $m_{\text{conj}} = M_{w_{\max}} - m_{\max}$
    $\mathbf{c}_w^{\mathbf{r}_k} \leftarrow$`singleDictionaryResidualUpdate(`$m_{\text{conj}}$, w, $\mathbf{c}_w^{\mathbf{r}_k}$`)`
  end
end

Function $\mathbf{c}_w \leftarrow$`singleDictionaryResidualUpdate(`$m_{\max}$, w, $\mathbf{c}_w$`)`**:**
  $a_{\text{rat}} \leftarrow a_w / a_{w_{\max}}$, $M_{\text{rat}} \leftarrow M_{w_{\max}} / M_w$
  $a_{\text{step}} = a_{\text{rat}}$ or 1 if $a_{\text{rat}} < 1$, $M_{\text{step}} = M_{\text{rat}}$ or 1 if $M_{\text{rat}} < 1$
  `/* Determine index sets */`
  Define $\mathcal{M}$, horizontal index set with stride M_{step} in cross kernel $\mathbf{h}_{w_{\max},w}$ taking into the account the misalignment of the grids.
  Define $\mathcal{N}$, vertical index set in a similar way using stride a_{step}.
  Define $\mathcal{I}$, residual coefficient vector index set covered by the kernel.
  `/* Update the residual (as in (18)) using truncated, subsampled and modulated cross-kernel (17): */`
  $\mathbf{c}_w(\mathcal{I}) = \mathbf{c}_w(\mathcal{I}) - c_{\max} \mathbf{h}_{w_{\max},w}^{(m_{\max},n_{\max})}(\mathcal{M}, \mathcal{N})$
return

Algorithm 2: Step 2c: Coefficient-domain MP residual update

proposed implementation to sliding local MP [51] and real-time MP mentioned in [21]. We will also investigate the structure of the Gram matrix of a dictionary consisting of an ensemble of real-valued windowed discrete cosine bases used by Ravelli et. al. [25] to determine whether similar acceleration technique is feasible.

ACKNOWLEDGMENT

The author wishes to thank Nicki Holighaus and Peter Balazs for discussions, proof-reading and valuable comments. The author also thanks Bob L. Sturm for sharing his thoughts on the subject in a form of a blog *Pursuits in the Null Space*.

REFERENCES

[1] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Constructive Approximation*, vol. 13, no. 1, pp. 57–98, 1997.
[2] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Tran. Signal Processing*, vol. 41, no. 12, pp. 3397–3415, Dec 1993.
[3] R. A. DeVore and V. N. Temlyakov, "Some remarks on greedy algorithms," *Advances in Computational Mathematics*, vol. 5, no. 1, pp. 173–187, Dec 1996.
[4] R. Gribonval, R. M. Figueras i Ventura, and P. Vandergheynst, "A simple test to check the optimality of a sparse signal approximation," *Signal Processing*, vol. 86, no. 3, pp. 496 – 510, 2006.
[5] R. Gribonval and P. Vandergheynst, "On the exponential convergence of matching pursuits in quasi-incoherent dictionaries," *IEEE Tran. Information Theory*, vol. 52, no. 1, pp. 255–261, Jan 2006.
[6] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Asilomar Conference on Signals, Systems and Computers*, Nov 1993, pp. 40–44 vol.1.
[7] G. M. Davis, S. G. Mallat, and Z. Zhang, "Adaptive time-frequency decompositions," *Optical Engineering*, vol. 33, no. 7, pp. 2183–2191, 1994.
[8] G. Rath and C. Guillemot, "A complementary matching pursuit algorithm for sparse approximation," in *Proc. 16th European Signal Processing Conference (EUSIPCO)*, Aug 2008, pp. 1–5.
[9] ——, "On a simple derivation of the complementary matching pursuit," *Signal Processing*, vol. 90, no. 2, pp. 702 – 706, 2010.
[10] B. L. Sturm and M. G. Christensen, "Cyclic matching pursuits with multiscale time-frequency dictionaries," in *Conf. Record of the 44th Asilomar Conference on Signals, Systems and Computers*, Nov 2010, pp. 581–585.
[11] B. L. Sturm, M. G. Christensen, and R. Gribonval, "Cyclic pure greedy algorithms for recovering compressively sampled sparse signals," in *Conference Record of the 45th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2011, pp. 1143–1147.
[12] T. Blumensath and M. E. Davies, "Gradient pursuit for non-linear sparse signal modelling," in *Proc. European Signal Processing Conference (EUSIPCO)*. IEEE, 2008, pp. 1–5.
[13] ——, "Gradient pursuits," *IEEE Tran. Signal Processing*, vol. 56, no. 6, pp. 2370–2382, 2008.
[14] B. Mailhé, R. Gribonval, F. Bimbot, and P. Vandergheynst, "A low complexity orthogonal matching pursuit for sparse signal approximation with shift-invariant dictionaries," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICAASP)*, April 2009, pp. 3445–3448.
[15] B. Mailhé, R. Gribonval, P. Vandergheynst, and F. Bimbot, "Fast orthogonal sparse approximation algorithms over local dictionaries," *Signal Processing*, vol. 91, no. 12, pp. 2822 – 2835, 2011.
[16] L. Rebollo-Neira, M. Rozložník, and P. Sasmal, "Analysis of a low memory implementation of the orthogonal matching pursuit greedy strategy," *CoRR*, vol. abs/1609.00053v2, 2017.
[17] M. Yaghoobi and M. E. Davies, "Fast and scalable: A survey on sparse approximation methods," The University of Edinburgh, Tech. Rep., 2009.
[18] I. Rish and G. Grabarnik, *Sparse modeling: Theory, algorithms, and applications*. CRC Press, 2015.
[19] B. L. Sturm, "Sparse approximation and atomic decomposition: Considering atom interactions in evaluating and building signal representations," Ph.D. dissertation, University of California, 2009.
[20] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, "Sparse representations in audio and music: From coding to source separation," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 995–1005, June 2010.
[21] D. Zantalis, "Guided matching pursuit and its application to sound source separation," Ph.D. dissertation, University of York, 2016.
[22] R. Gribonval, P. Depalle, X. Rodet, E. Bacry, and S. Mallat, "Sound signals decomposition using a high resolution matching pursuit," in *Proc. Int. Computer Music Conf. (ICMC'96)*, Aug. 1996, pp. 293–296.
[23] R. Gribonval, "Fast matching pursuit with a multiscale dictionary of Gaussian chirps," *IEEE Tran. Signal Processing*, vol. 49, no. 5, pp. 994–1001, May 2001.
[24] B. L. Sturm and J. D. Gibson, "Matching pursuit decompositions of non-noisy speech signals using several dictionaries," in *Proc. IEEE Int.*

Conf. on Acoustics Speech and Signal Processing, vol. 3, May 2006, pp. III–III.

[25] E. Ravelli, G. Richard, and L. Daudet, "Union of MDCT bases for audio coding," *IEEE Tran. Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1361–1372, Nov 2008.

[26] G. Chardon, T. Necciari, and P. Balazs, "Perceptual matching pursuit with Gabor dictionaries and time-frequency masking," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 3102–3106.

[27] O. Derrien, "Time-scaling of audio signals with muti-scale Gabor analysis," in *Proc. Int. Conf. Digital Audio Effects (DAFx–07)*, Bordeaux, France, Sep. 2007.

[28] B. L. Sturm, L. Daudet, and C. Roads, "Pitch-shifting audio signals using sparse atomic approximations," in *Proc. 1st ACM Workshop on Audio and Music Computing Multimedia*, ser. AMCMM '06. New York, NY, USA: ACM, 2006, pp. 45–52.

[29] R. Gribonval, "Sparse decomposition of stereo signals with matching pursuit and application to blind separation of more than two sources from a stereo mixture," in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 3, May 2002, pp. III–3057–III–3060.

[30] G. Bhattacharya and P. Depalle, "Sparse denoising of audio by greedy time-frequency shrinkage," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 2898–2902.

[31] P. Leveau and L. Daudet, "Multi-resolution partial tracking with modified matching pursuit," in *Proc. 14th European Signal Processing Conference*, Sept 2006, pp. 1–4.

[32] P. Durka, *Matching pursuit and unification in EEG analysis*, ser. Artech House Engineering in Medicine and Biology Series. Artech House, Inc., 2007.

[33] S. Li, "Discrete multi-gabor expansions," *IEEE Tran. Information Theory*, vol. 45, no. 6, pp. 1954–1967, Sep 1999.

[34] P. J. Wolfe, S. J. Godsill, and M. Dörfler, "Multi-gabor dictionaries for audio time-frequency analysis," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2001, pp. 43–46.

[35] M. Zibulski and Y. Y. Zeevi, "Discrete multiwindow gabor-type transforms," *IEEE Tran. Signal Processing*, vol. 45, no. 6, pp. 1428–1442, Jun 1997.

[36] P. L. Søndergaard, B. Torrésani, and P. Balazs, "The linear time frequency analysis toolbox," *International Journal of Wavelets, Multiresolution Analysis and Information Processing*, vol. 10, no. 4, pp. 1–27, 2012.

[37] Z. Průša, P. L. Søndergaard, N. Holighaus, C. Wiesmeyr, and P. Balazs, "The large time-frequency analysis toolbox 2.0," in *Sound, Music, and Motion*, ser. LNCS. Springer International Publishing, 2014, pp. 419–442.

[38] S. E. Ferrando, L. A. Kolasa, and N. Kovačević, "Algorithm 820: A flexible implementation of matching pursuit for Gabor functions on the interval," *ACM Trans. Math. Softw.*, vol. 28, no. 3, pp. 337–353, Sep. 2002.

[39] S. Krstulović and R. Gribonval, "MPTK: Matching pursuit made tractable," in *Proc. Int. Conf. on Acoustics Speech and Signal Processing ICASSP 2006*, vol. 3, May 2006, pp. III–496–III–499.

[40] P. Balazs, "Frames and finite dimensionality: Frame transformation, classification and algorithms," *Applied Mathematical Sciences*, vol. 2, no. 41–44, pp. 2131–2144, 2008.

[41] D. F. Walnut, "Continuity properties of the gabor frame operator," *Journal of Mathematical Analysis and Applications*, vol. 165, no. 2, pp. 479 – 504, 1992.

[42] P. L. Søndergaard, "Efficient algorithms for the discrete Gabor transform with a long FIR window," *J. Fourier Anal. Appl.*, vol. 18, no. 3, pp. 456–470, 2012.

[43] S. Mallat, *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*, 3rd ed. Academic Press, 2008.

[44] K. Gröchenig, *Foundations of time-frequency analysis*, ser. Applied and numerical harmonic analysis. Boston, Basel, Berlin: Birkhäuser, 2001.

[45] J. Le Roux, H. Kameoka, N. Ono, and S. Sagayama, "Fast signal reconstruction from magnitude STFT spectrogram based on spectrogram consistency," in *Proc. 13th Int. Conf. on Digital Audio Effects (DAFx-10)*, Sep. 2010, pp. 397–403.

[46] M. R. Portnoff, "Implementation of the digital phase vocoder using the fast fourier transform," *IEEE Tran. Acoustics, Speech, and Signal Processing*, vol. 24, no. 3, pp. 243–248, Jun 1976.

[47] The European Broadcasting Union, "Tech 3253: Sound quality assessment material recordings for subjective tests," Geneva, Tech. Rep., Sept. 2008. [Online]. Available: https://tech.ebu.ch/docs/tech/tech3253.pdf

[48] M. Frigo and S. G. Johnson, "The design and implementation of FFTW3," *Proc. of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005, special issue on "Program Generation, Optimization, and Platform Adaptation".

[49] L. Daudet, "Sparse and structured decompositions of signals with the molecular matching pursuit," *IEEE Tran. Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1808–1816, Sept 2006.

[50] H. Lahdili, H. Najaf-Zadeh, R. Pichevar, and L. Thibault, "Perceptual matching pursuit for audio coding," in *Audio Engineering Society Convention 124*, May 2008.

[51] L. Daudet, "Audio sparse decompositions in parallel: Let the greed be shared!" *IEEE Signal Processing Magazine*, vol. 27, no. 2, pp. 90–96, March 2010.