Acoustic Image Estimation using Fast Transforms

Vítor H. Nascimento
University of São Paulo
São Paulo, Brazil
vitor@lps.usp.br

Mateus C. Silva
University of São Paulo
São Paulo, Brazil

Bruno S. Masiero
University of Campinas
Campinas, Brazil
http://www.decom.fee.unicamp.br/~masiero

***Abstract*—The estimation of acoustic images is a computationally-intensive task for large microphone arrays. One of the most popular algorithms for acoustic image estimation, delay-and-sum beamforming (DAS), has low computational complexity, but low spatial resolution. Several methods have been developed to obtain higher resolution, among which compressive beamforming, which is based on sparse estimation. Although this method does achieve higher resolution, its computational complexity is significantly higher than that of DAS. In this paper we describe the use of the Kronecker array transform (KAT) to accelerate DAS and specific algorithms for sparse estimation, in particular matching pursuit (MP) and the spectral projected-gradient algorithm SPGL1. In addition, we describe how the nonequispaced fast Fourier transform (NFFT) can be used to provide further acceleration.**

I. Introduction

Microphone arrays are widely used to map noise sources, for example in the aircraft industry [1], [2], through the construction of acoustic images. These are maps of the sound intensities impinging on a microphone array as a function of the direction of arrival [3]. In order to keep computational complexity low, acoustic images are usually estimated using beamforming algorithms, in particular delay-and-sum (DAS) [4], [5], which linearly combines delayed versions of the signals arriving at each microphone, so that signals coming from the direction of interest are in phase. In order to form an acoustic image, the procedure must be repeated for each direction of interest.

Despite its simplicity, a problem with this technique is its low spatial resolution due to sidelobes. Different approaches were propose to solve this problem, for example, deconvolution methods such as DAMAS and DAMAS2 [6], [7], as well as covariance-fitting methods, which rely on regularized optimization algorithms [4], [8]. Although these algorithms give good results, they require that the sound sources be uncorrelated and are computationally intensive, particularly for large arrays and high image resolutions. This former constraint (uncorrelated sources) was removed in the DAMAS-C algorithm and in the covariance-matrix fitting (CMF) method of [4], but the problem of computational complexity largely remained.

With the aim of reducing the computational complexity for these methods, [8], [9] introduced the Kronecker Array Transform (KAT), essentially a way of taking advantage of the structure that arises in the problem when the array is planar and the microphones are placed on a rectangular grid ("separable" arrays). The original KAT of [9] was designed for methods based on the autocovariance matrix of the Fourier transform of the microphone signals (such as DAMAS2 and CMF) [8]. Additionally, [9] described how the nonequispaced fast Fourier transform (NFFT) [10] can be used (alone or in conjunction with the KAT) to further accelerate the computations. The restriction to separable arrays excludes the application of the KAT with arrays with more complex structures, such as multi-arm logarithm spiral arrays [11]. However, [8] shows that, when using either deconvolution or CMF methods, the performance obtained with non-uniform separable arrays is comparable to that obtained with spiral arrays.

More recently, [12] introduced the compressive beamforming algorithm, which uses regularized optimization techniques (using the ℓ_1 norm) to avoid using the autocorrelation matrix of the microphone signals, and thus eliminates the necessity of uncorrelated sources. A generalized version of the KAT was proposed for compressive beamforming in [13], allowing substantial reduction in computational complexity, as explained in more detail in the next section. Note that the generalized KAT can also be used to accelerate standard DAS computations, under the restriction of using a separable array.

In this paper we describe how the NFFT can be used to accelerate computations for DAS and compressive beamforming, alone or combined with the generalized KAT, and estimate computational gains when different optimization algorithms (such as Matching Pursuit — MP [14] — and the spectral projected-gradient algorithm known as SPGL1 [15]) are used for solving the sparse estimation problems.

II. Imaging Algorithms

We assume we have an array with M microphones placed at positions $\boldsymbol{p}_m \in \mathbb{R}^3$. We are interested in estimating the signals (or signal powers) at a frequency ω_k arriving from a set of directions $\boldsymbol{u}_n$. The directions are parametrized in $\boldsymbol{u}$-space [5], that is, we assume that ($(\cdot)^T$ denotes transposition)

$$\boldsymbol{u}_n = \begin{bmatrix} u_{x,n} & u_{y,n} & \sqrt{1 - u_{x,n}^2, d - u_{y,n}^2} \end{bmatrix}^T, \tag{1}$$

where $0 \leq u_{x,n}^2 + u_{y,n}^2 \leq 1$.

A. Signal model

Let $\boldsymbol{x}(\omega_k) = [x_0(\omega_k) \quad \ldots \quad x_{M-1}(\omega_k)]^T$ be a vector collecting the discrete Fourier transforms (DFT) of each microphone at a single frequency ω_k, computed from a length-B block of the time-domain signals arriving at the array. We approximate the sound field at the array by a superposition of plane waves, each one arriving from one of the chosen look directions $\mathbf{u}_n$. In this case, $\boldsymbol{x}(\omega_k)$ is given by [3], [16]

$$\boldsymbol{x}(\omega_k) = \boldsymbol{V}(\omega_k)\, \boldsymbol{y}(\omega_k) + \boldsymbol{\eta}(\omega_k), \tag{2}$$

where $\boldsymbol{y}(\omega_k) = [y_0(\omega_k) \;\; y_1(\omega_k) \;\; \cdots \;\; y_{N-1}(\omega_k)]^T$ represents the source signals in the frequency domain and $\boldsymbol{\eta}(\omega_k)$ represents frequency-domain noise. The array manifold matrix $\boldsymbol{V}(\omega_k) = [\boldsymbol{v}(\mathbf{u}_0, \omega_k) \;\; \boldsymbol{v}(\mathbf{u}_1, \omega_k) \;\; \cdots \;\; \boldsymbol{v}(\mathbf{u}_{N-1}, \omega_k)]$, of size $M \times N$, describes the transfer function between the source arriving from $\mathbf{u}_n$ and the sensor at $\mathbf{p}_m$, at frequency ω_k.

Assuming that the sources are in the far field, the array manifold vector for source n is then given by

$$\boldsymbol{v}(\mathbf{u}_n, \omega_k) = \left[e^{j(\omega_k/c)\mathbf{u}_n^T \mathbf{p}_0} \;\; \ldots \;\; e^{j(\omega_k/c)\mathbf{u}_n^T \mathbf{p}_{M-1}} \right]^T . \tag{3}$$

Please note that in the remaining of this manuscript we omit the dependency on ω_k in $\boldsymbol{V}(\omega_k)$, $\boldsymbol{y}(\omega_k)$ and $\boldsymbol{x}(\omega_k)$ in order to simplify notation.

B. Spatial filtering: Conventional beamformer

Our problem is to estimate the $N \times 1$ vector $\boldsymbol{y}$ from the $M \times 1$ vector $\boldsymbol{x}$, in which in general $M \ll N$. Due to its simplicity, the most commonly applied methods are the delay-and-sum (DAS) beamformer [5], [17], [18], which can be mathematically modeled as a weighted sum of the signals captured by the microphones, i.e., $\widehat{y} = \boldsymbol{w}^H \boldsymbol{x}$, where $\boldsymbol{w} = [w_1 \;\; \cdots \;\; w_M]^T$ is a complex weight vector. DAS compensates for the relative delay at each sensor for waves arriving from a given direction, and averages the aligned signals:

$$\boldsymbol{w}_{\text{DAS}}(\boldsymbol{u}) = \frac{1}{M}\boldsymbol{v}(\boldsymbol{u}) = \frac{\boldsymbol{v}(\boldsymbol{u})}{\boldsymbol{v}^H(\boldsymbol{u})\boldsymbol{v}(\boldsymbol{u})}. \tag{4}$$

To obtain the acoustic image, a different beamformer is applied for each direction $\boldsymbol{u}_n$ in our grid. This is equivalent to

$$\hat{\boldsymbol{y}} = \frac{1}{M}\boldsymbol{V}^H \boldsymbol{x}, \tag{5}$$

and the sound intensities are approximated either directly by $|[\hat{\boldsymbol{y}}]_n|^2$, where $[\hat{\boldsymbol{y}}]_n$ denotes the n-th entry of $\hat{\boldsymbol{y}}$, or by an average computed from several signal blocks.

C. Covariance-matrix fitting (CMF)

In CMF methods, we first estimate the autocovariance matrix of the microphone signals, $\boldsymbol{S} = \mathrm{E}\{\boldsymbol{x}\boldsymbol{x}^H\}$,from several blocks of the input signals. From (2),

$$\boldsymbol{S} = \boldsymbol{V}\,\mathrm{E}\{\boldsymbol{y}\boldsymbol{y}^H\}\boldsymbol{V}^H + \mathrm{E}\{\boldsymbol{\eta}\boldsymbol{\eta}^H\}. \tag{6}$$

Assuming that the sources are uncorrelated ([4] relaxes this condition), the source autocovariance matrix $\mathrm{E}\{\boldsymbol{y}\boldsymbol{y}^H\} = \mathrm{diag}(Y_n)$ is diagonal, so we can write [9]

$$\boldsymbol{S} = \sum_{n=0}^{N-1} Y_n \boldsymbol{v}(\boldsymbol{u}_n)\boldsymbol{v}^H(\boldsymbol{u}_n) + \mathrm{E}\{\boldsymbol{\eta}\boldsymbol{\eta}^H\}, \tag{7}$$

where Y_n are the pixels of the acoustic image we wish to estimate. Applying the vec operator (which stacks the columns of a matrix one at the top of the other) to (7) and using Kronecker product properties [19], we obtain [9]

$$\mathrm{vec}(\boldsymbol{S}) = \boldsymbol{A}\,\mathrm{col}(Y_n) + \mathrm{vec}\left(\mathrm{E}\{\boldsymbol{\eta}\boldsymbol{\eta}^H\}\right). \tag{8}$$

Defining $\boldsymbol{s} = \mathrm{vec}(\boldsymbol{S})$, $\boldsymbol{z} = \mathrm{col}(Y_n)$, $\boldsymbol{\sigma} = \mathrm{vec}\left(\mathrm{E}\{\boldsymbol{\eta}\boldsymbol{\eta}^H\}\right)$ and

$$\boldsymbol{A} \triangleq [\boldsymbol{v}^*(\boldsymbol{u}_0) \otimes \boldsymbol{v}(\boldsymbol{u}_0) \quad \ldots \quad \boldsymbol{v}^*(\boldsymbol{u}_{N-1}) \otimes \boldsymbol{v}(\boldsymbol{u}_{N-1})],$$

we obtain

$$\boldsymbol{s} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{\sigma}, \tag{9}$$

where $\otimes$ denotes the Kronecker product, $(\cdot)^*$ denotes complex conjugate, and matrix $\boldsymbol{A}$ is $M^2 \times N$.

The covariance-matrix fitting method consists of finding a solution $\widehat{\boldsymbol{z}}$ to (9) through a regularized optimization problem. The solution proposed in [4] uses ℓ_1 regularization:

$$\widehat{\boldsymbol{z}} = \arg\min_{\boldsymbol{z}} \|\boldsymbol{A}\boldsymbol{z} - \boldsymbol{s}\|_2^2 \text{ s.t. } \|\boldsymbol{z}\|_1 \leq \beta, \tag{10}$$

where β is a regularization constant. This approach is adequate for sparse images. [8] uses a slightly modified optimization problem, in order to use the SPGL1 package [15]

$$\widehat{\boldsymbol{z}} = \arg\min_{\boldsymbol{z}} \|\boldsymbol{z}\|_1, \text{ s.t. } \|\boldsymbol{A}\boldsymbol{z} - \boldsymbol{s}\|_2^2 \leq \sigma_0^2, \tag{11}$$

where the regularization term σ_0^2 depends on an estimate of $\|\boldsymbol{\sigma}\|_2^2$ (related to the noise variance). Another possibility described in [8] is the use of total variation (TV) regularization, which would be adequate for smooth images. Although our results can also be applied to this case, due to page restrictions we do not consider TV regularization in this paper.

D. Compressive beamforming

CMF techniques work well, but the introduction of the autocovariance matrix results in large matrices (such as in (9)), and the treatment of correlated sources is cumbersome. Beamforming techniques are robust to noise and fast, but suffer from low resolution and the presence of sidelobes [4], [8], [12]. To counter these effects, a recent work proposes to estimate directly $\boldsymbol{y}$ (instead of the image $\boldsymbol{z}$) as a ℓ_1-norm minimization problem [12], that is, solving

$$\hat{\boldsymbol{y}} = \arg\min_{\boldsymbol{y}} \|\boldsymbol{y}\|_1, \text{ s.t. } \|\boldsymbol{V}\boldsymbol{y} - \boldsymbol{x}\|_2^2 \leq \sigma_1^2, \tag{12}$$

where again σ_1^2 is a regularization term, whose choice depends on the noise variance.

We describe in the next section the Kronecker array transform, which was introduced to accelerate computations required for the solution of regularized optimization problems such as (11) and (12). These optimization problems are solved iteratively [14], [15], and for large arrays, the calculation bottleneck usually lies in matrix-vector products such as $\boldsymbol{A}\boldsymbol{z}$ and $\boldsymbol{A}^H\boldsymbol{s}$ (for (11)) or $\boldsymbol{V}\boldsymbol{y}$ and $\boldsymbol{V}^H\boldsymbol{x}$ (for (12)) [8]. The original KAT was designed to accelerate the solution of CMF problems involving $\boldsymbol{A}$ [9]. More recently, the KAT was extended to compressive beamforming problems such as (12) [13].

III. Kronecker Array Transform

The KAT is based on decomposing matrices $\boldsymbol{V}$ and $\boldsymbol{A}$ in smaller matrices using Kronecker products. The Kronecker product was already used to reduce the complexity in the solution of certain linear systems of equations in [20], [21]. The KAT, on the other hand, accelerates matrix-vector computations.

The KAT requires an array in which the M microphones are placed on a planar, rectangular grid. Assuming without loss of generality that the array is placed horizontally, this means that the microphone coordinates $\mathbf{p}_0, \cdots, \mathbf{p}_{M-1} \in \mathbb{R}^3$ satisfy

$$\mathbf{p}_{\ell+iM_y} = [p_x(\ell) \quad p_y(i) \quad 0]^T, \tag{13}$$

where $0 \le \ell \le M_x - 1$, $0 \le i \le M_y - 1$, $M = M_x M_y$ and $p_x(\ell)$ and $p_y(i)$ are the x- and y-coordinates.

The look directions must also be chosen following a rectangular grid, that is, we want to estimate the power of the signals arriving at the array from directions $\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{N-1}$, such that

$$\boldsymbol{u}_{q+rN_y} = \left[u_x(q) \quad u_y(r) \quad \sqrt{1 - u_x^2(q) - u_y^2(r)}\right]^T, \tag{14}$$

where $-1 \le u_x(q), u_y(r) \le 1$, $0 \le q \le N_x - 1$, $0 \le r \le N_y - 1$ and $N = N_x N_y$ is the total number of look directions in which we are interested. Note that only directions with $u_x^2(q) + u_y^2(r) \le 1$ make physical sense.

A. Kronecker factorization

When both the microphone positions $\mathbf{p}_m$ and the look directions $\mathbf{u}_n$ are chosen in the form of rectangular grids (i.e., they satisfy (13) and (14)), [13] shows that the array manifold matrix $\boldsymbol{V}$ can be factored in terms of smaller matrices $\boldsymbol{V}_x$ and $\boldsymbol{V}_y$ using the Kronecker product [19].

$$\boldsymbol{V} = \boldsymbol{V}_x \otimes \boldsymbol{V}_y, \tag{15}$$

where the (ℓ, q)-th element of $\boldsymbol{V}_x$ and the (i, r)-th element of $\boldsymbol{V}_y$ are

$$\boldsymbol{V}_x[\ell, q] = e^{j\omega_k u_x(q) p_x(\ell)/c}, \tag{16a}$$

$$\boldsymbol{V}_y[i, r] = e^{j\omega_k u_y(r) p_y(i)/c}. \tag{16b}$$

Relation (15) follows directly from substituting (13) and (14) in the definition of the (m, n)-th element of $\boldsymbol{V}$

$$\boldsymbol{V}[m, n] = e^{j\omega_k \mathbf{u}_n^T \mathbf{p}_m/c} = e^{j\omega_k u_x(q) p_x(\ell)/c} e^{j\omega_k u_y(r) p_y(i)/c}, \tag{17}$$

where $m = \ell + iM_x$, $n = q + rN_x$.

A similar decomposition is described in [9] for matrix $\boldsymbol{A}$. Since application of the original KAT for CMF problems is thoroughly described in [8], we concentrate here on the application of the generalized KAT described in [13] to the compressive beamforming problem (12). The results in Sections III-B–III-D are proven in [13].

B. Direct Fast Transform

Consider a matrix-vector product of the form $\widehat{\boldsymbol{x}} = \boldsymbol{V}\widehat{\boldsymbol{y}}$. Directly computing the product requires $4MN$ real multiplications and $4MN - 2M$ real additions. On the other hand, substituting (15) results in

$$\widehat{\boldsymbol{x}} = (\boldsymbol{V}_x \otimes \boldsymbol{V}_y)\, \widehat{\boldsymbol{y}}. \tag{18}$$

Using the well-known Kronecker product identity [19]

$$\operatorname{vec}(\boldsymbol{D}\boldsymbol{Z}\boldsymbol{C}^T) = (\boldsymbol{C} \otimes \boldsymbol{D}) \operatorname{vec}(\boldsymbol{Z}), \tag{19}$$

we rewrite (18) as

$$\widehat{\boldsymbol{X}} = \boldsymbol{V}_y \widehat{\boldsymbol{Y}} \boldsymbol{V}_x^T, \tag{20}$$

where $\widehat{\boldsymbol{x}} = \operatorname{vec}(\widehat{\boldsymbol{X}})$ and $\widehat{\boldsymbol{y}} = \operatorname{vec}(\widehat{\boldsymbol{Y}})$. The output matrix $\widehat{\boldsymbol{X}} \in \mathbb{C}^{M_y \times M_x}$ contains the values of $\widehat{\boldsymbol{x}}$ arranged in the same geometrical disposition as the sensors in the array, with the columns of the matrix representing the vertical y-axis and the rows of the matrix representing the horizontal x-axis. The same is valid for the signal matrix $\widehat{\boldsymbol{Y}} \in \mathbb{C}^{N_y \times N_x}$, that contains all values of $\widehat{\boldsymbol{y}}$ arranged in the same geometrical disposition as the scan grid. The total number of operations now depends on the order the operations are performed. Computing $\widehat{\boldsymbol{Y}}\boldsymbol{V}_x^T$ requires $4M_x N_x N_y = 4M_x N$ real multiplications and $4M_x N - 2M_x N_y$ real additions. The computation of $\boldsymbol{V}_y \left(\widehat{\boldsymbol{Y}}\boldsymbol{V}_x^T\right)$ requires $4MN_y$ real multiplications, and $4MN_y - 2M_x$ real additions, so the total number of operations required for (20) is $4M_x N + 4MN_y$ real multiplications, and $4M_x N + 4MN_y - 2M_x N_y - 2M_x$ real additions if we compute $\left(\widehat{\boldsymbol{Y}}\boldsymbol{V}_x^T\right)$ first. Using $\left(\boldsymbol{V}_y \widehat{\boldsymbol{Y}}\right)\boldsymbol{V}_x^T$ requires $4M_y N + 4MN_x$ real multiplications, and $4M_y N + 4MN_x - 2M_y N_x - 2M_y$ real additions. Assuming $M_x = M_y$ and $N_x = N_y$, both options have the same complexity, and the gain in number of operations (considering only multiplications) is

$$\gamma_{\text{KAT}} = \frac{4MN}{4M^{1/2}N + 4MN^{1/2}} \ge \frac{1}{2}\min\{M^{1/2}, N^{1/2}\}. \tag{21}$$

C. Adjoint Fast Transform

To speed-up the calculation of DAS (5) we can apply (15) to the adjoint matrix-vector product $\widehat{\boldsymbol{y}} = \boldsymbol{V}^H \boldsymbol{x}$, resulting into

$$\widehat{\boldsymbol{y}} = (\boldsymbol{V}_x \otimes \boldsymbol{V}_y)^H \boldsymbol{x}, \tag{22}$$

which can also be rewritten in a fast form using identity (19), such that

$$\widehat{\boldsymbol{Y}} = \boldsymbol{V}_y^H \boldsymbol{X} \boldsymbol{V}_x^*. \tag{23}$$

Note that this has the same computational cost as the direct transform.

D. Direct-Adjoint Fast Transform

Combining (18) and (22) we write the direct-adjoint matrix-vector product as

$$\widehat{\boldsymbol{y}} = \boldsymbol{V}^H \boldsymbol{V} \boldsymbol{y} = [(\boldsymbol{V}_x^H \boldsymbol{V}_x) \otimes (\boldsymbol{V}_y^H \boldsymbol{V}_y)]\boldsymbol{y}. \tag{24}$$

Now, applying identity (19), we obtain the fast transform form given by

$$\widehat{\boldsymbol{Y}} = \boldsymbol{V}_y^H \boldsymbol{V}_y \boldsymbol{Y} \boldsymbol{V}_x^T \boldsymbol{V}_x^*. \tag{25}$$

This implementation is especially interesting when M_x and M_y are sufficiently large in comparison to N_x and N_y, because it can be evaluated as $\widehat{\boldsymbol{Y}} = (\boldsymbol{V}_y^H \boldsymbol{V}_y)\boldsymbol{Y}(\boldsymbol{V}_x^T \boldsymbol{V}_x^*)$, with precomputed versions of $(\boldsymbol{V}_y^H \boldsymbol{V}_y)$ and $(\boldsymbol{V}_x^T \boldsymbol{V}_x^*)$.

IV. Further acceleration with the NFFT

The non-equispaced fast Fourier transform (NFFT) is an (approximate) method for efficient evaluation of sums of the form [10]

$$c(g_1, g_2) = \sum_{h_1, h_2 \in I_H} d(h_1, h_2) e^{-j2\pi(h_1 w_{1,g_1} + h_2 w_{2,g_2})}, \tag{26}$$

where $g_1, g_2 = 0, 1, \ldots, G-1$, I_H is a set of H integers in the interval $\left[-\frac{N_H}{2}, \frac{N_H}{2}\right)$, $N_H \ge H$ is an even positive integer, the w_{g_1}, w_{g_2} are real numbers, and the $d(h_1, h_2)$ are complex.

Note that the elements of I_H need not be uniformly spaced. The NFFT was used in [9] to accelerate the matrix-vector products that appear in the CMF methods. Here we extend this result to matrix-vector products involving matrix $\boldsymbol{V}$, by itself or in conjunction with the KAT. The computational complexity depends on the desired accuracy ϵ, and is of the order of $\mathcal{O}(H \log H + |\log \epsilon|^d G)$, where $d = 2$ for the case of two-dimensional NFFTs as in (26), or $d = 1$ for one-dimensional NFFTs. An algorithm with the same asymptotic complexity is available for the adjoint problem

$$\hat{d}(h_1, h_2) = \sum_{g_1, g_2 = 0}^{G-1} c(g_1, g_2) e^{j2\pi(h_1 w_{1,g_1} + h_2 w_{2,g_2})}. \qquad (27)$$

Consider first the application of the NFFT directly to a matrix-vector product $\boldsymbol{V}\boldsymbol{y}$. Consider element (m, n) from $\boldsymbol{V}$ (17). If we choose the $u_x(q)$ and $u_y(r)$ as

$$u_x(q) = \frac{2}{N_x}\left(-\frac{N_x}{2} + q\right), \quad u_y(r) = \frac{2}{N_y}\left(-\frac{N_y}{2} + r\right),$$

and define

$$w_{1,\ell} = -\frac{\omega_k}{2\pi c}\frac{N_x}{2}p_x(\ell), \qquad w_{2,i} = -\frac{\omega_k}{2\pi c}\frac{N_y}{2}p_y(i),$$

then the product $\boldsymbol{V}\boldsymbol{y}$ is in the form of a two-dimensional NFFT. The adjoint NFFT (27) can be used to accelerate the computation of $\boldsymbol{V}^H\boldsymbol{x}$. The complexity is of order $\mathcal{O}(N \log N + |\log \epsilon|^2 M)$.

The NFFT can also be used in conjunction with the KAT. In this case, when computing the product $\boldsymbol{V}_y\boldsymbol{Y}\boldsymbol{V}_x^T$, each row of $\boldsymbol{Y}\boldsymbol{V}_x^T$ corresponds to a one-dimensional NFFT, with complexity $\mathcal{O}(N_x \log N_x + |\log \epsilon| M_x)$. Considering that there are M_x such rows, and repeating the procedure for each column of $\boldsymbol{V}_y\left(\boldsymbol{Y}\boldsymbol{V}_x^T\right)$, the total complexity results $\mathcal{O}\left(N \log N_x + M_x N_y \log N_y + |\log \epsilon|(M_x N_y + M)\right)$.

V. Examples of application

In this section we give a few examples of application of the KAT, first directly to DAS, and later to two algorithms used to solve sparse approximation problems: SPGL1 [15] and matching pursuit (MP) [14].

A. DAS

DAS computations can be accelerated directly by using the adjoint KAT, the adjoint NFFT or both in conjunction, since DAS corresponds simply to the operation $\widehat{\boldsymbol{y}} = \boldsymbol{V}^H\boldsymbol{x}$. The reduction in computational complexity is therefore given by (21). Fig. 1a shows an example of DAS in a case with five sources, $M_x = M_y = 8$, $N_x = N_y = 64$.

B. Matching pursuit

The matching pursuit algorithm [14, Table 1] is a method for finding sparse solutions to linear systems of equations such as $\boldsymbol{V}\boldsymbol{y} \approx \boldsymbol{x}$. In its initialization, we must evaluate $\boldsymbol{V}^H\boldsymbol{x}$ and $\boldsymbol{V}^H\boldsymbol{V}$, which can be accelerated with the KAT and/or the NFFT. The main loop contains only length-M scalar-vector products and vector additions, which are relatively low-cost and cannot be accelerated using either method. Without using either method, the total number of real multiplications is $4N^2M + 4NM + 6NN_{\text{it}}$, where N_{it} is the number of iterations. Using the KAT and assuming $N_x = N_y$ and $M_x = M_y$, the cost reduces to $4N^{3/2}M + 4N^2M^{1/2} + N^{1/2}M + NM^{1/2} + 6NN_{\text{it}}$ multiplications (the figures for additions are similar). For $M = 64$ and $N = 64^2$ and assuming 600 iterations, this results a gain of 7.8 (approximately $M^{1/2}$) less computations. Fig. 1b shows the same example as in Fig. 1a, but using MP.

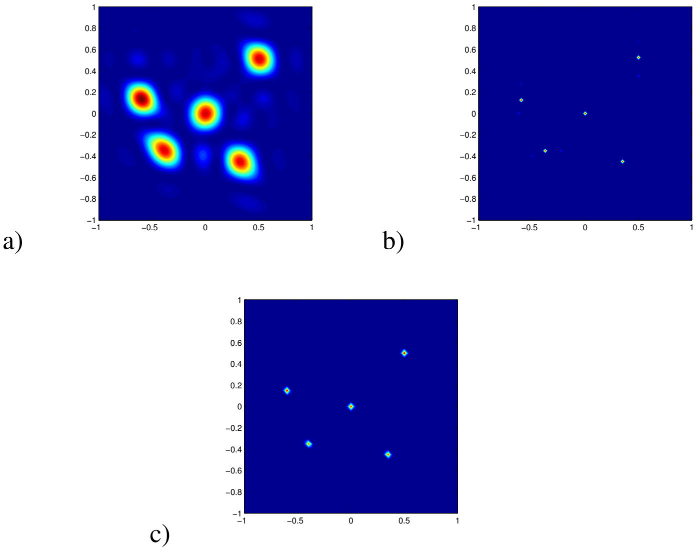

Fig. 1. Acoustic image examples, $M_x = M_y = 8$, $N_x = N_y = 64$. Top left, DAS; top right, MP; bottom, SPGL1.

C. SPGL1

SPGL1 is an advanced method for solving (12) [15]. The main loop of the algorithm requires one matrix-vector product by $\boldsymbol{V}$ and one matrix-vector product by $\boldsymbol{V}^H$. Together with a projection operation requiring $\mathcal{O}(\mathcal{N} \log \mathcal{N})$ operations, these are the most computational-intensive operations in the algorithms. The reduction in computational complexity obtained by using the KAT is expected to be of the order of $M^{1/2}$ (assuming $M \ll N$). An example is shown in Fig. 1c.

VI. Conclusion

In this paper we described ways of accelerating algorithms for estimation of acoustic images using a recently-proposed version of the Kronecker array transform, applied to DAS beamforming and to compressive beamforming. For compressive beamforming, we described the computational gains to be expected when using two different algorithms for the solution of the sparse estimation problem: matching pursuit, a low-complexity greedy algorithm, and the more advanced SPGL1 package. In addition, we described how the computations can be further accelerated using the nonequispaced FFT.

Acknowledgment

This work was partly supported by FAPESP (Grants 14/06066-6 and 14/04256-2) and CNPq (Grant 306268/2014-0).

References

[1] S. Lee, "Phased-array measurement of modern regional aircraft turbofan engine noise," *AIAA Journal*, vol. 2653, 2006.

[2] W. Humphreys and T. Brooks, "Noise spectra and directivity for a scale-model landing gear," *International Journal of Aeroacoustics*, vol. 8, no. 5, pp. 409–443, 2009.

[3] V. H. Nascimento, B. S. Masiero, and F. P. Ribeiro, "Acoustic imaging using the Kronecker array transform," in *Signals and Images: Advances and Results in Speech, Estimation, Compression, Recognition, Filtering, and Processing*, R. F. Coelho, V. H. Nascimento, R. L. de Queiroz, J. M. T. Romano, and C. C. Cavalcante, Eds. CRC Press, 2015, ch. 6, pp. 153–178.

[4] T. Yardibi, J. Li, P. Stoica, N. S. Zawodny, and L. N. C. Iii, "A covariance fitting approach for correlated acoustic source mapping," *The Journal of the Acoustical Society of America*, vol. 127, no. 5, pp. 2920–2931, May 2010, 00014.

[5] H. L. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*. New York, NY: John Wiley & Sons, 2002.

[6] R. Dougherty, "Extensions of DAMAS and benefits and limitations of deconvolution in beamforming," *AIAA paper*, vol. 2961, no. 11, 2005.

[7] T. F. Brooks and W. M. Humphreys, "A deconvolution approach for the mapping of acoustic sources (DAMAS) determined from phased microphone arrays," *Journal of Sound and Vibration*, vol. 294, no. 4, pp. 856–879, 2006.

[8] F. P. Ribeiro and V. H. Nascimento, "Computationally efficient regularized acoustic imaging," in *Proc. IEEE Int Acoustics, Speech and Signal Processing Conf. (ICASSP)*, 2011, pp. 2688–2691.

[9] ——, "Fast transforms for acoustic imaging— part I: Theory," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2229–2240, 2011.

[10] J. Keiner, S. Kunis, and D. Potts, "Using NFFT 3—a software library for various nonequispaced fast Fourier transforms," *ACM Transactions on Mathematical Software (TOMS)*, vol. 36, no. 4, p. 19, 2009.

[11] J. R. Underbrink and R. P. Dougherty, "Array design for non-intrusive measurement of noise sources," *NOISE-CON 96*, pp. 757–762, 1996.

[12] A. Xenaki, P. Gerstoft, and K. Mosegaard, "Compressive beamforming." *The Journal of the Acoustical Society of America*, vol. 136, no. 1, pp. 260–271, 2014. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/24993212

[13] B. S. Masiero and V. H. Nascimento, "Kronecker array transform," 2016, submitted to the IEEE Signal Process. Lett. [Online]. Available: http://www.lps.usp.br/vitor/artigos/KhatriRao_v4.pdf

[14] Y. V. Zakharov and V. H. Nascimento, "Orthogonal matching pursuit with DCD iterations," *Electronics Letters*, vol. 49, no. 4, pp. 295–297, 2013.

[15] E. van den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2008.

[16] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing: Concepts and Techniques*. Prentice Hall, Englewood-Cliffs N.J., 1993.

[17] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE Signal Process. Mag.*, vol. 13, no. 4, pp. 67–94, 1996.

[18] T. Yardibi, J. Li, P. Stoica, and L. Cattafesta III, "Sparsity constrained deconvolution approaches for acoustic source mapping," *The Journal of the Acoustical Society of America*, vol. 123, p. 2631, 2008.

[19] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, MA: Cambridge University Press, 1991.

[20] C. Van Loan and N. Pitsianis, "Approximation with Kronecker products," in *Linear Algebra for Large Scale and Real Time Applications*, 1993, no. 1991, pp. 293–314.

[21] H. Lev-Ari, T. Kailath, and J. Cioffi, "Least-squares adaptive lattice and transversal filters: A unified geometric theory," *IEEE Trans. Inf. Theory*, vol. 30, no. 2, pp. 222–236, 1984.