# Fast Transforms for Acoustic Imaging—Part I: Theory

Flávio P. Ribeiro, *Student Member, IEEE*, and Vítor H. Nascimento, *Member, IEEE*

***Abstract*—The classical approach for acoustic imaging consists of beamforming, and produces the source distribution of interest convolved with the array point spread function. This convolution smears the image of interest, significantly reducing its effective resolution. Deconvolution methods have been proposed to enhance acoustic images and have produced significant improvements. Other proposals involve covariance fitting techniques, which avoid deconvolution altogether. However, in their traditional presentation, these enhanced reconstruction methods have very high computational costs, mostly because they have no means of efficiently transforming back and forth between a hypothetical image and the measured data. In this paper, we propose the Kronecker Array Transform (KAT), a fast separable transform for array imaging applications. Under the assumption of a separable array, it enables the acceleration of imaging techniques by several orders of magnitude with respect to the fastest previously available methods, and enables the use of state-of-the-art regularized least-squares solvers. Using the KAT, one can reconstruct images with higher resolutions than was previously possible and use more accurate reconstruction techniques, opening new and exciting possibilities for acoustic imaging.**

***Index Terms*—Acoustic imaging, array imaging, array processing, fast transform, regularized least squares, sparse reconstruction.**

## I. Introduction

ACOUSTIC imaging refers to the problem of mapping the locations and intensities of sound sources over a region of interest using microphone arrays. For example, a microphone array can be positioned in a wind tunnel to determine the noise distribution over a model due to high velocity airflow [1], [2]. These measurements are routinely used to design cars, trains, and aircraft, which are quieter to outside observers and to passengers. Microphone arrays have been employed to measure the noise generated by turbofan engines [3] and wind turbines [4] for similar noise reduction applications. Acoustic imaging has also been used to visualize the reverberant structure of concert halls [5]. We note that techniques for imaging the shapes of objects and structures via acoustic waves are sometimes called acoustic imaging, but we will not address these problems.

Manuscript received May 26, 2010; revised December 21, 2010; accepted February 09, 2011. Date of publication February 22, 2011; date of current version July 15, 2011. This work was supported in part by the São Paulo Research Foundation (FAPESP) and in part by the National Council for Scientific and Technological Development (CNPq). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. John P. Kerekes.

The authors are with the Electronic Systems Engineering Department, Escola Politécnica, Universidade de São Paulo, São Paulo, SP 05508-900, Brazil (e-mail: fr@lps.usp.br; vitor@lps.usp.br).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2011.2118220

Array imaging is possible because sensor arrays can be electronically steered toward arbitrary directions. One can define a grid over a region of interest, electronically steer the array over all elements of the grid, and, thus, create a map of estimated sound pressure levels. Each point in the grid can be represented as a pixel. The value of the pixel can be chosen to represent the estimated sound pressure level, thus creating an acoustic image.

Array imaging differs from source localization techniques, such as [6]–[10], because these usually produce a pseudospectrum of the wavefield, with maxima that indicate the location of dominant sources but with values that do not map to source powers. Thus, in this paper, we will assume that accurate power estimates are desirable.

The simplest and most common method for imaging uses delay and sum beamforming. This technique consists of delaying and summing the signals arriving at each sensor so that the sources located at a direction of interest are reinforced, and sources located in other directions are attenuated. Beamforming is simple, but unfortunately produces the lowest quality images. Indeed, under the assumption that the sources are in the far-field of the array, beamforming produces the source distribution of interest convolved with the array point spread function (PSF). Since a typical acoustic array has a relatively small aperture with respect to its operating wavelengths, its PSF can be quite large, so that delay and sum beamforming produces very smeared images. Alternative beamforming techniques have been developed to improve resolution by using data-dependent methods and numerical optimization of the beampattern [11]–[13], but they do not overcome the fundamental limitation that beamforming produces convolved images.

To overcome this limitation, several deconvolution techniques have been proposed [14]–[17]. They use as inputs the image obtained with delay and sum beamforming and the array PSF, and generally produce a much better approximation of the original source distribution. Nevertheless, deconvolution is an ill-conditioned inverse problem, and typically requires some knowledge of the solution to discriminate between different solutions which would be equally good fits for the measured data. The acoustic imaging methods proposed so far tend to use very simple types of regularization, such as low-pass filtering between iterations, or no regularization at all.

Regularized signal reconstruction has been a topic of interest for many decades, and gained significant momentum with the popularity of compressive sensing [18]–[20]. Indeed, many image reconstruction problems can be recast as convex optimization problems, which can be solved with computationally efficient iterative methods. While many of these techniques were designed for imaging applications, they have remained limited to fields, such as medical image reconstruction. Therefore, most of these developments have not been applied to acoustic imaging.

1057-7149/$26.00 © 2011 IEEE

A major reason for this separation between fields has been the absence of computationally efficient transforms for aeroacoustic imaging. For example, consider the generic nonlinear signal reconstruction problem given by

$$\hat{\mathbf{y}} = \arg\min_{\mathbf{y}} ||\mathbf{\Psi y}|| \text{ such that } \mathbf{\Phi y} = \mathbf{x} \tag{1}$$

where $\mathbf{x}$ is the measured signal, $\hat{\mathbf{y}}$ is the reconstructed signal, $\mathbf{\Psi}$ is a sparsifying transform, and $\mathbf{\Phi}$ is a transform which models the measurement process. For an acoustic image, $\mathbf{y}$ would be a vectorized version of the image describing the true source distribution, and $\mathbf{x}$ would be a vectorized version of the array's sample covariance matrix.

Since, in practice, (1) is solved iteratively, one must be able to quickly evaluate $\mathbf{\Psi u}$ and $\mathbf{\Phi u}$ (and $\mathbf{\Psi}^H\mathbf{v}$ and $\mathbf{\Phi}^H\mathbf{v}$, as we will see in [21, Sec. II] of this paper) for arbitrary $\mathbf{u}, \mathbf{v}$. This is a very strong requirement, because the application of these transforms is the bottleneck of efficient convex optimization algorithms and completely determines their computational costs (regardless of whether the transform is fast or slow). While one can choose a convenient fast sparsifying transform $\mathbf{\Psi}$, the transform $\mathbf{\Phi}$ is determined by the physical measurement process. For example, for MRI applications, we naturally have $\mathbf{\Phi} = \mathbf{P}\mathcal{F}$, where $\mathcal{F}$ is a fast Fourier transform (FFT) and $\mathbf{P}$ is a subsampling operator. Finite differences have been successfully used as the sparsifying transform $\mathbf{\Psi}$ with $\|\cdot\| = \|\cdot\|_1$ [22], [23].

While sparsity-enforcing approaches have been proposed for the direction of arrival estimation [24] and acoustic imaging [25], to our knowledge, no method of acoustic imaging uses a fast implementation of $\mathbf{\Phi}$. To motivate the need for a fast transform, consider a naive matrix representation of $\mathbf{\Phi}$. Given an array of $N$ sensors and an image with $M$ pixels, $\mathbf{\Phi}$ has $N^2$ rows and $M$ columns. For $N^2 = M = 256^2$, $\mathbf{\Phi}$ has 4 billion elements and the products $\mathbf{\Phi u}$ and $\mathbf{\Phi}^H\mathbf{v}$ are computationally very expensive, making convex optimization methods intractable with current desktop computers. Thus, the naive implementation of $\mathbf{\Phi}$ is only practical for very small images and arrays, thus motivating the development of a fast transform.

In this paper, we develop the Kronecker Array Transform (KAT), a fast transform which implements $\mathbf{\Phi}$, $\mathbf{\Phi}^H$, and $\mathbf{\Phi}^H\mathbf{\Phi}$ for separable arrays. The KAT can be applied to many existing array imaging algorithms, with significant performance gains. It also allows for the use of state-of-the-art solvers for acoustic imaging problems, obviating ad-hoc solutions which typically produce worse results. Indeed, with a fast transform, one can use most of the general purpose, state-of-the-art solvers developed for other imaging and compressive sensing applications. By combining the KAT with these methods, we can accelerate reconstruction times by several orders of magnitude with respect to the fastest previously available implementations. In practical terms, an image which would take minutes to reconstruct can be obtained in a few seconds. Finally, while this transform was motivated by applications in aeroacoustics, it also applies to generic wave fields and separable sensor arrays.

To our knowledge, previous proposals for accelerated acoustic imaging are all based on beamforming. Zimmerman and Studer [26] propose offloading delay-and-sum beamforming to a field-programmable gate array (FPGA), which performs all of the computation and draws acoustic images over a framebuffer. While this approach makes beamforming faster, it does not reduce its underlying computational cost. Huang [27] uses a state observer model to recursively obtain an approximation of the acoustic image while acquiring data (in contrast to computing an image from a sample covariance matrix). While this method has the advantage of returning incremental results, it has the same computational cost as beamforming, and a comparable beampattern. In contrast, the KAT dramatically reduces the underlying computational costs of acoustic imaging, allowing more accurate reconstruction methods to be used instead of beamforming.

Part I is organized as follows: Section II gives several definitions and further motivates the need for fast transforms. Section III introduces the KAT, its adjoint and its direct-adjoint composition, under the assumption of far-field sources. Section IV presents connections with the fast Fourier transform, fast non-equispaced Fourier transform (NFFT), and fast non-equispaced in time and frequency Fourier transform (NNFFT) [28]. These connections are also a contribution, because to our knowledge, the NFFT and NNFFT have never been used for acoustic imaging. We show how the NFFT and NNFFT can also be used to accelerate acoustic imaging under a far-field approximation, despite being an order of magnitude slower than the KAT. Section V presents benchmarks comparing the KAT with the NFFT, NNFFT, and explicit matrix representations. Section VI shows how to extend the KAT for near-field imaging, modeling spherical wavefronts instead of planar wavefronts. This generalization is unique to our proposal, and produces a transform which is orders of magnitude faster than direct matrix multiplication (which becomes the only alternative, since the FFT, NFFT, and NNFFT require a far-field approximation). Section VII concludes this paper.

Reference [21, Sec. II] presents applications. Section I briefly reviews the results from Part I. In Section II, we use the KAT to significantly accelerate existing techniques and to enable the use of general-purpose solvers, obtaining more accurate reconstructions than possible with current state-of-the-art methods. Section III features examples and compares several reconstruction methods with respect to computational cost and accuracy. Section IV compares separable arrays with multiarm logarithmic spiral arrays, and shows that by requiring separable arrays, we are not trading reconstruction quality for speed. Finally, Section V consists of conclusions and final comments.

## II. Preliminaries

Consider a sensor array composed of $N$ microphones at Cartesian coordinates $\mathbf{p}_0, \ldots, \mathbf{p}_{N-1} \in \mathbb{R}^3$, and an arbitrary wavefield which we wish to estimate. Suppose that this wave field can be modeled as generated by the superposition of $M$ point sources located at coordinates $\mathbf{q}_0, \ldots, \mathbf{q}_{M-1} \in \mathbb{R}^3$, where $M$ may be a large number in order to obtain an accurate model. Let

$$\mathbf{Q} = [\mathbf{q}_0 \quad \mathbf{q}_1 \quad \cdots \quad \mathbf{q}_{M-1}]. \tag{2}$$

The time-domain samples of each microphone are segmented into frames of $K$ samples, and each frame is converted to the

TABLE I
LIST OF SYMBOLS

| Symbol | Meaning |
|---|---|
| $^T$ | matrix or vector transpose |
| $^H$ | matrix or vector Hermitian transpose |
| $^*$ | complex conjugate |
| $\mathrm{mod}\,(a,b)$ | remainder of $a/b$, for $a,b \in \mathbb{Z}_+$ |
| $\lfloor x \rfloor$ | round-off of $x \in \mathbb{R}$ towards $-\infty$ |
| $\mathrm{vec}\,\{\cdot\}$ | vectorization operator |
| $\otimes$ | Kronecker product |
| $\odot$ | Hadamard product |
| $M = M_x M_y$ | number of image pixels |
| $N = N_x N_y$ | number of array microphones |
| $\omega$ | operating frequency in the Fourier domain |
| $\mathbf{q}_m$ | source coordinates, for $0 \le m < M$ |
| $\mathbf{u}_m$ | source coordinates in U-space, for $0 \le m < M$ |
| $\mathbf{p}_n$ | array element coordinates, for $0 \le n < N$ |
| $\mathbf{v}_{\mathbf{u}_m}(\omega)$ | $N \times 1$ array manifold vector for look direction $\mathbf{u}_m$ |
| $\mathbf{v}_{x_n}(\omega)$ | $N_x \times 1$ separable component of $\mathbf{v}_{\mathbf{u}_m}(\omega)$ |
| $\mathbf{v}_{y_n}(\omega)$ | $N_y \times 1$ separable component of $\mathbf{v}_{\mathbf{u}_m}(\omega)$ |
| $\mathrm{v}^i_{x_n}(\omega)$ | $i^{\mathrm{th}}$ element of $\mathbf{v}_{x_n}(\omega)$ |
| $\mathbf{Y}(\omega)$ | $M_x \times M_y$ image at a frequency $\omega$ |
| $\mathbf{S}(\omega)$ | $N \times N$ spectral matrix at a frequency $\omega$ |
| $\mathbf{B}_{i,j}$ | $(i,j)$ entry of matrix $\mathbf{B}$ |
| $\mathbb{1}$ | matrix of ones |

frequency domain using a fast Fourier transform (FFT). In the presence of additive noise, the $N \times 1$ array output vector for a single frequency on a single frame can be modeled as

$$\mathbf{x}(\omega_k) = \mathbf{V}(\mathbf{Q}, \omega_k)\mathbf{f}(\omega_k) + \boldsymbol{\eta}(\omega_k), \tag{3}$$

where $0 \le k < K/2$, $\mathbf{V}(\mathbf{Q}, \omega_k) = [\mathbf{v}(\mathbf{q}_0, \omega_k)\mathbf{v}(\mathbf{q}_1, \omega_k) \cdots \mathbf{v}(\mathbf{q}_{M-1}, \omega_k)]$ is the array manifold matrix, $\mathbf{f}(\omega_k) = [f_0(\omega_k) f_1(\omega_k) \cdots f_{M-1}(\omega_k)]^T$ is the frequency-domain signal waveform, and $\boldsymbol{\eta}(\omega_k)$ is the frequency-domain noise waveform.

The near-field array manifold vector for source $m$ is given by [29]

$$\mathbf{v}(\mathbf{q}_m, \omega_k) = \left[ \frac{\mathrm{e}^{-j\frac{\omega_k}{c}\|\mathbf{p}_0 - \mathbf{q}_m\|}}{\|\mathbf{p}_0 - \mathbf{q}_m\|} \quad \cdots \quad \frac{\mathrm{e}^{-j\frac{\omega_k}{c}\|\mathbf{p}_{N-1} - \mathbf{q}_m\|}}{\|\mathbf{p}_{N-1} - \mathbf{q}_m\|} \right]^T \tag{4}$$

where $c$ is the speed of sound.

Define $\mathbf{u}_m = \mathbf{q}_m / \|\mathbf{q}_m\|$, the look direction for source $m$. Under a far-field approximation (modeling a plane wave), the time differences of arrival is given by $-\mathbf{u}_m^T \mathbf{p}_n / c$, for $0 \le n < N$. Since the wavefront is not expanding, the $1/\|\mathbf{p}_n - \mathbf{q}_m\|$ attenuation disappears, and the far-field array manifold vector for source $m$ is given by

$$\mathbf{v}(\mathbf{u}_m, \omega_k) = \left[ \mathrm{e}^{j\frac{\omega_k}{c}\mathbf{u}_m^T \mathbf{p}_0} \quad \cdots \quad \mathrm{e}^{j\frac{\omega_k}{c}\mathbf{u}_m^T \mathbf{p}_{N-1}} \right]^T. \tag{5}$$

Using spherical coordinates

$$\mathbf{u}_m = \begin{bmatrix} \sin\phi\cos\theta \\ \sin\phi\sin\theta \\ \cos\phi \end{bmatrix} \tag{6}$$

where $\theta$ and $\phi$ are the azimuth and elevation angles, respectively. One can reparameterize the unit half-sphere by defining

$$u_x(\theta, \phi) = \sin\phi\cos\theta \tag{7}$$

$$u_y(\theta, \phi) = \sin\phi\sin\theta \tag{8}$$

so that

$$\mathbf{u}_m = \begin{bmatrix} u_x \\ u_y \\ \sqrt{1 - u_x^2 - u_y^2} \end{bmatrix} \tag{9}$$

for $u_x^2 + u_y^2 \le 1$. Uniform sampling in U-space (where $\mathrm{U} = [-1, 1]^2$) is convenient in many applications, because under a far-field approximation, it makes point-spread functions shift-invariant. In this paper, it will enable us to decouple the $x$ and $y$ axes, producing the fast transform.

Since the optimizations presented in the following sections require Cartesian (not necessarily uniform) parameterizations in U-space and far-field approximations, we will assume that manifold vectors have the form (5). In Section VI, we extend our results for near-field sources, which will allow us to approximate (4) with arbitrary accuracy.

Using the assumption of far-field sources, we rewrite (3)as

$$\mathbf{x}(\omega_k) = \mathbf{V}(\omega_k)\mathbf{f}(\omega_k) + \boldsymbol{\eta}(\omega_k) \tag{10}$$

where $\mathbf{V}(\omega_k) = [\mathbf{v}(\mathbf{u}_0, \omega_k)\ \mathbf{v}(\mathbf{u}_1, \omega_k)\ \cdots\ \mathbf{v}(\mathbf{u}_{M-1}, \omega_k)]$.

Let

$$\mathbf{S}_{\mathbf{x}}(\omega_k) = \mathrm{E}\left\{\mathbf{x}(\omega_k)\mathbf{x}^H(\omega_k)\right\} \tag{11}$$

be the array's narrowband cross spectral matrix for $0 \le k < K/2$. If $\mathbf{x}_0(\omega_k), \ldots, \mathbf{x}_{L-1}(\omega_k)$ corresponds to $L$ frequency-domain frames (also known as snapshots), the spectral matrix can be estimated with

$$\mathbf{S}_{\mathbf{x}}(\omega_k) = \frac{1}{L}\sum_{l=0}^{L-1} \mathbf{x}_l(\omega_k)\mathbf{x}_l^H(\omega_k). \tag{12}$$

We assume that the statistics of the signal and noise are stationary over the measured period, so that (12) is an unbiased estimator.

Processing $\mathbf{S}_{\mathbf{x}}(\omega_k)$ instead of each $\mathbf{x}_l(\omega_k)$ is typically more convenient, because $\mathbf{S}_{\mathbf{x}}(\omega_k)$ carries only the relative phase shifts between microphones and is the result of averaging, so that it has less noise content. Indeed, for each $0 \le l < L$, $\mathbf{x}_l(\omega_k)$ has a phase shift which is equal for every element but unknown, which disappears when computing $\mathbf{S}_{\mathbf{x}}(\omega_k)$. To save space, in the following text, we will assume narrowband processing and omit the argument $\omega_k$. Also, the $\mathbf{x}$ subscript will be dropped, and $\mathbf{S}_{\mathbf{x}}(\omega_k)$ will be written as $\mathbf{S}$.

Substituting (10) into (11) and assuming that the noise is spatially white and uncorrelated with the sources of interest, we have

$$\mathbf{S}_{\mathbf{x}}(\omega_k) = \mathbf{V}(\omega_k)\mathrm{E}\left\{\mathbf{f}(\omega_k)\mathbf{f}^H(\omega_k)\right\}\mathbf{V}^H(\omega_k) + \sigma^2\mathbf{I} \tag{13}$$

where $\sigma^2 = \mathrm{E}\{\eta_i(\omega_k)\eta_i^*(\omega_k)\}$, $0 \le i < N$.

Assume that the wavefield impinging on the array can be modeled as emitted by the superposition of uncorrelated point sources located in the array's far field. One can represent these sources by a collection of points at coordinates $\{\mathbf{u}_i\}_{i=0}^{M-1}$ located in a sufficiently fine grid in U-space. This representation is effectively a 2-D digital image, where the pixel coordinates

correspond to locations in U-space, and the pixel values correspond to source intensities. Note that in (13), assuming that the sources are uncorrelated implies that $\mathrm{E}\{\mathbf{f}(\omega_k)\mathbf{f}^H(\omega_k)\}$ is diagonal. Furthermore, the diagonal of $\mathrm{E}\{\mathbf{f}(\omega_k)\mathbf{f}^H(\omega_k)\}$ is a vectorized version of the acoustic image.

Given an acoustic image where each pixel corresponds to a point source, one can easily obtain the array spectral matrix as long as all point sources (pixels) are assumed to be pairwise uncorrelated. If there were cross-correlations, one would drop the assumption that $\mathrm{E}\{\mathbf{f}(\omega_k)\mathbf{f}^H(\omega_k)\}$ is diagonal, and require the correlation coefficient for each pair of sources, whose determination would be clearly impractical even for small images (for example, a $64 \times 64$ pixel image would have $64^2 = 4096$ pixels (sources) and $(1/2)4096 \cdot 4097 \approx 8$ million unique cross-correlations). Therefore, unless stated otherwise, we shall assume that sources are pairwise uncorrelated.

However, we note that a fast transform can also be obtained for correlated source distributions as a natural generalization of the KAT. It has special importance because one cannot apply Fourier methods to accelerate the reconstruction of correlated source maps. Nevertheless, since the number of cross-correlations scales quadratically, estimating every cross term is only viable for very simple source distributions. Thus, a proposal for imaging correlated sources should combine a fast transform and domain-specific regularization, the latter being an open problem. Due to space limitations, we will not address these topics in this text.

Recall that to solve (1) efficiently, one requires a fast method of obtaining $\mathbf{S}$ from a hypothetical image. Consider an $M_x \times M_y$ pixel acoustic image, define $M = M_x M_y$ and let $\mathbf{u}_0, \ldots, \mathbf{u}_{M-1}$ be an enumeration of all pixel coordinates in U-space. Let $\mathbf{v}(\mathbf{u}_m)$ be the array manifold vector when steered towards the look direction $\mathbf{u}_m$. For a single source at $\mathbf{u}_m$ radiating with power $|Y(\mathbf{u}_m)|^2$, the measured spectral matrix is $\mathbf{S} = |Y(\mathbf{u}_m)|^2 \mathbf{v}(\mathbf{u}_m)\mathbf{v}^H(\mathbf{u}_m)$. Given the source powers for $\mathbf{u}_0, \ldots, \mathbf{u}_{M-1}$, one can reconstruct $\mathbf{S}$ by superposition, so that

$$\mathbf{S} = \sum_{m=0}^{M-1} |Y(\mathbf{u}_m)|^2 \mathbf{v}(\mathbf{u}_m)\mathbf{v}^H(\mathbf{u}_m). \tag{14}$$

Unless the image is very sparse, this expression becomes computationally intractable. For instance, consider a 256 element array and a $256 \times 256$ acoustic image. Each $|Y(\mathbf{u}_m)|^2 \mathbf{v}(\mathbf{u}_m)\mathbf{v}^H(\mathbf{u}_m)$ outer product generates a $256\times 256$ matrix. Neglecting the cost to scale $\mathbf{v}(\mathbf{u}_m)$ by $|Y(\mathbf{u}_m)|$, the outer product requires $2^{16}$ complex multiply-accumulate (MAC) instructions.[1] This process must be repeated $M = 2^{16}$ times, resulting in $2^{32}$ complex MACs. Since each outer product has Hermitian symmetry, it suffices to determine its upper or lower triangular part (including the main diagonal), which reduces the total complex MAC count to approximately $2^{31}$. Nevertheless, this computational cost is still excessive for a transform intended to be used in an iterative method. In the following text, we describe how to implement an efficient transform to obtain $\mathbf{S}$ from $\{|Y(\mathbf{u}_m)|^2\}_{0 \le m < M}$.

[1]Modern digital-signal-processing (DSP) architectures are able to implement a multiplication followed by an accumulation in the same clock cycle. This single cycle instruction is known as a MAC. Since the computational cost of performing a sum, product, or MAC is the same, for the purposes of estimating computational complexity, it suffices to estimate the total number of MACs.

## III. Kronecker Array Transform

Define $\mathbf{y} = [|Y(\mathbf{u}_0)|^2 \ \cdots \ |Y(\mathbf{u}_{M-1})|^2]^T$. Let us write (14) as a linear transform $\mathbf{A}$ so that $\mathbf{s} = \mathbf{A}\mathbf{y}$, with $\mathbf{s} = \mathrm{vec}\{\mathbf{S}\}$. To save space, we will write $\mathbf{v}(\mathbf{u}_m)$ as $\mathbf{v}_{\mathbf{u}_m}$, and will denote its $i$th element by $\mathrm{v}^i_{u_m}$ (elements of array manifold vectors will be indexed using superscripts). Let $N$ be the number of microphones in the array. Note that

$$\mathrm{vec}\left\{\mathbf{v}_{\mathbf{u}_m}\mathbf{v}^H_{\mathbf{u}_m}\right\} = \begin{bmatrix} \mathbf{v}_{\mathbf{u}_m}\mathrm{v}^{0*}_{\mathbf{u}_m} \\ \mathbf{v}_{\mathbf{u}_m}\mathrm{v}^{1*}_{\mathbf{u}_m} \\ \vdots \\ \mathbf{v}_{\mathbf{u}_m}\mathrm{v}^{(N-1)*}_{\mathbf{u}_m} \end{bmatrix}. \tag{15}$$

Therefore

$$\begin{aligned} \mathbf{s} &= \mathbf{A}\mathbf{y} \qquad (16)\\ &= \begin{bmatrix} \mathbf{v}_{\mathbf{u}_0}\mathrm{v}^{0*}_{\mathbf{u}_0} & \mathbf{v}_{\mathbf{u}_1}\mathrm{v}^{0*}_{\mathbf{u}_1} & \cdots & \mathbf{v}_{\mathbf{u}_{M-1}}\mathrm{v}^{0*}_{\mathbf{u}_{M-1}} \\ \mathbf{v}_{\mathbf{u}_0}\mathrm{v}^{1*}_{\mathbf{u}_0} & \mathbf{v}_{\mathbf{u}_1}\mathrm{v}^{1*}_{\mathbf{u}_1} & \cdots & \mathbf{v}_{\mathbf{u}_{M-1}}\mathrm{v}^{1*}_{\mathbf{u}_{M-1}} \\ \vdots & \vdots & & \vdots \\ \mathbf{v}_{\mathbf{u}_0}\mathrm{v}^{(N-1)*}_{\mathbf{u}_0} & \mathbf{v}_{\mathbf{u}_1}\mathrm{v}^{(N-1)*}_{\mathbf{u}_1} & \cdots & \mathbf{v}_{\mathbf{u}_{M-1}}\mathrm{v}^{(N-1)*}_{\mathbf{u}_{M-1}} \end{bmatrix}\mathbf{y} \\ &= \begin{bmatrix} \mathbf{v}^*_{\mathbf{u}_0} \otimes \mathbf{v}_{\mathbf{u}_0} & \mathbf{v}^*_{\mathbf{u}_1} \otimes \mathbf{v}_{\mathbf{u}_1} & \cdots \\ & & \mathbf{v}^*_{\mathbf{u}_{M-1}} \otimes \mathbf{v}_{\mathbf{u}_{M-1}} \end{bmatrix}\mathbf{y}, \qquad (17) \end{aligned}$$

where $\otimes$ is the Kronecker product.

Given a 2-D array, its array manifold vector $\mathbf{v}(\mathbf{u}) = \mathbf{v}(u_x, u_y)$ is said to be separable if $\mathbf{a}(u_x)$ exists and $\mathbf{b}(u_x)$ so that $\mathbf{v}(u_x, u_y) = \mathbf{a}(u_x) \otimes \mathbf{b}(u_y)$ for all valid $u_x, u_y$. Note that $\mathbf{a}(u_x)$ and $\mathbf{b}(u_y)$ need not be submanifold vectors. We say that an array is separable if and only if it has a separable manifold vector. We will show below how the array geometry relates to its separability under a far-field assumption.

To simplify the notation that follows, let us specify the enumeration $\mathbf{u}_0, \ldots, \mathbf{u}_{M-1}$ of look directions we are using. Suppose that $\mathbf{Y}$ is a $M_x \times M_y$ digital image representing the acoustic image. The rows of $\mathbf{Y}$ correspond to horizontal scan lines of arbitrarily sampled pixels, and the columns of $\mathbf{Y}$ correspond to vertical scan lines of arbitrarily sampled pixels. Let $\{u_{x_m}\}_{0 \le m < M_x}$ and $\{u_{y_n}\}_{0 \le n < M_y}$ be points which sample the U-space along the $x$ and $y$ axes, ordered from left to right and from top to bottom. We define $\mathbf{u}_0, \ldots, \mathbf{u}_{M-1}$ so that

$$\mathbf{y} = \mathrm{vec}\{\mathbf{Y}\} = \begin{bmatrix} |Y(\mathbf{u}_0)|^2 \\ |Y(\mathbf{u}_1)|^2 \\ \vdots \\ |Y(\mathbf{u}_{M-1})|^2 \end{bmatrix}. \tag{18}$$

Breaking $\mathbf{u}$ into components, this implies that

$$\mathbf{u}_m = \begin{bmatrix} u_{x_{\lfloor m/M_y \rfloor}} \\ u_{y_{\mathrm{mod}(m, M_y)}} \\ \sqrt{1 - \left(u_{x_{\lfloor m/M_y \rfloor}}\right)^2 - \left(u_{y_{\mathrm{mod}(m, M_y)}}\right)^2} \end{bmatrix}. \tag{19}$$

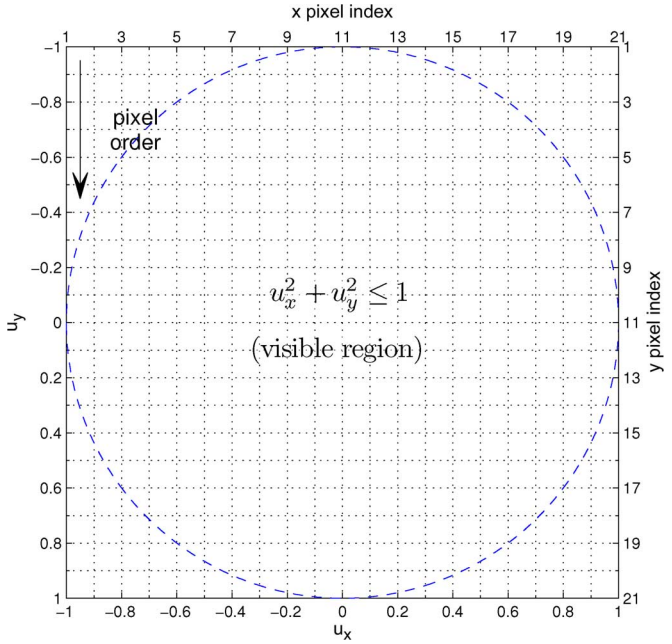


Fig. 1. Example of pixel order and U-space parameterization for an acoustic image, for $M_x = M_y = 21$ and uniform sampling in the U-space.

Fig. 1 shows an example of how pixels are ordered and parameterized in U-space.

We now show that under the far-field parameterization given by (5), an array is separable if and only if it has elements positioned over a (potentially nonuniform) Cartesian grid.

To see this, consider an array with sensor coordinates $\mathbf{p}_i \in \mathbb{R}^3$, for $0 \leq i < N$, with $x$ and $y$ coordinates drawn from $\{p_{x_i}\}_{i=0}^{N_x-1}$ and $\{p_{y_i}\}_{i=0}^{N_y-1}$, respectively, so that

$$\mathbf{p}_i = \begin{bmatrix} p_{x_{\lfloor i/N_y \rfloor}} & p_{y_{\mathrm{mod}(i,N_y)}} & 0 \end{bmatrix}^T.$$

Let $N = N_x N_y$ be the number of array elements. Define a horizontal array with sensor coordinates $\mathbf{p}_{x_i} \in \mathbb{R}^3$, for $0 \leq i < N_x$ and a vertical array with sensor coordinates $\mathbf{p}_{y_j} \in \mathbb{R}^3$, for $0 \leq j < N_y$, so that

$$\mathbf{p}_{x_i} = [p_{x_i} \quad 0 \quad 0]^T \qquad \mathbf{p}_{y_j} = [0 \quad p_{y_j} \quad 0]^T.$$

Let $\mathbf{v}_x$ and $\mathbf{v}_y$ be the $N_x \times 1$ and $N_y \times 1$ manifold vectors for these 1-D arrays. Then, for $0 \leq i < N_x$ and $0 \leq j < N_y$

$$\begin{aligned}
&[\mathbf{v}(u_x,u_y)]_{i\cdot N_y+j} \\
&\quad = \mathrm{e}^{j\frac{\omega_k}{c}\left[u_x \;\; u_y \;\; \sqrt{1-u_x^2-u_y^2}\right]\mathbf{p}_{i\cdot N_y+j}} \\
&\quad = \mathrm{e}^{j\frac{\omega_k}{c}\left[u_x \;\; u_y \;\; \sqrt{1-u_x^2-u_y^2}\right](\mathbf{p}_{x_i}+\mathbf{p}_{y_j})} \\
&\quad = \mathrm{e}^{j\frac{\omega_k}{c}\left[u_x \;\; 0 \;\; \sqrt{1-u_x^2}\right]\mathbf{p}_{x_i}} \mathrm{e}^{j\frac{\omega_k}{c}\left[0 \;\; u_y \;\; \sqrt{1-u_y^2}\right]\mathbf{p}_{y_j}} \\
&\quad = [\mathbf{v}_x(u_x)]_i\,[\mathbf{v}_y(u_y)]_j
\end{aligned}$$

which, by definition, is equivalent to

$$\mathbf{v}(u_x,u_y) = \mathbf{v}_x(u_x) \otimes \mathbf{v}_y(u_y). \tag{20}$$

Thus, arrays with Cartesian geometries are separable under U-space parameterization. To prove the converse, note that

$$\begin{aligned}
[\mathbf{v}(u_x,u_y)]_i &= \mathrm{e}^{j\frac{\omega_k}{c}\left[u_x \;\; u_y \;\; \sqrt{1-u_x^2-u_y^2}\right]\mathbf{p}_i} \\
&= \mathrm{e}^{j\frac{\omega_k}{c}\left(u_x\bar{p}_{x_i}+u_y\bar{p}_{y_i}\right)} \\
&= \mathrm{e}^{j\frac{\omega_k}{c}u_x\bar{p}_{x_i}}\,\mathrm{e}^{j\frac{\omega_k}{c}u_y\bar{p}_{y_i}}
\end{aligned} \tag{21}$$

where $\mathbf{p}_i = [\bar{p}_{x_i} \; \bar{p}_{y_i} \; 0]^T$. By hypothesis, $\mathbf{a}(u_x)$ and $\mathbf{b}(u_x)$ exist so that $\mathbf{v}(u_x,u_y) = \mathbf{a}(u_x) \otimes \mathbf{b}(u_y)$. The term $u_x\bar{p}_{x_i}$ from (21) must belong to $\mathbf{a}(u_x)$, since it is a function of $u_x$, and $\bar{p}_{x_i}$ is constant. It follows that $\mathbf{a}(u_x) = \mathbf{v}_x(u_x)$ and $\mathbf{b}(u_y) = \mathbf{v}_y(u_y)$, with $\mathbf{v}_x(u_x)$ and $\mathbf{v}_y(u_y)$ defined before, imply a Cartesian geometry.

### A. Fast Direct Transform

To save space, we will use the shorthand notation

$$\begin{aligned}
\mathbf{v}_x(u_{x_m}) &= \mathbf{v}_{x_m} = \left[\mathrm{v}_{x_m}^0 \;\; \mathrm{v}_{x_m}^1 \;\; \cdots \;\; \mathrm{v}_{x_m}^{N_x-1}\right]^T \\
\mathbf{v}_y(u_{y_n}) &= \mathbf{v}_{y_n} = \left[\mathrm{v}_{y_n}^0 \;\; \mathrm{v}_{y_n}^1 \;\; \cdots \;\; \mathrm{v}_{y_n}^{N_y-1}\right]^T.
\end{aligned} \tag{22}$$

Using the separability of the array in (17), we obtain

$$\begin{aligned}
\mathbf{A} = \big[&(\mathbf{v}_{x_0}^* \otimes \mathbf{v}_{y_0}^*) \otimes (\mathbf{v}_{x_0} \otimes \mathbf{v}_{y_0}) \;\; \cdots \\
&\cdots \;\; \left(\mathbf{v}_{x_{M-1}}^* \otimes \mathbf{v}_{y_{M-1}}^*\right) \otimes \left(\mathbf{v}_{x_{M-1}} \otimes \mathbf{v}_{y_{M-1}}\right)\big].
\end{aligned} \tag{23}$$

For $0 \leq m,n < N_xN_y$, the separability of the array also allows row $m \cdot N_xN_y + n$ of $\mathbf{A}$ to be written as

$$\left[\mathrm{v}_{x_0}^{i*}\mathrm{v}_{x_0}^{j} \;\; \cdots \;\; \mathrm{v}_{x_{M_x-1}}^{i*}\mathrm{v}_{x_{M_x-1}}^{j}\right] \otimes \left[\mathrm{v}_{y_0}^{k*}\mathrm{v}_{y_0}^{l} \;\; \cdots \;\; \mathrm{v}_{y_{M_y-1}}^{k*}\mathrm{v}_{y_{M_y-1}}^{l}\right] \tag{24}$$

where $i = \lfloor m/N_y \rfloor$, $j = \lfloor n/N_y \rfloor$, $k = \mathrm{mod}(m,N_y)$, and $l = \mathrm{mod}(n,N_y)$.

For $0 \leq i,j < N_x$ and $0 \leq k,l < N_y$, define

$$c_m(i,j) = \mathrm{v}_{x_m}^{i*}\mathrm{v}_{x_m}^{j} \qquad d_n(k,l) = \mathrm{v}_{y_n}^{k*}\mathrm{v}_{y_n}^{l}. \tag{25}$$

For $0 \leq m,n < N_xN_y$, an arbitrary element $\mathbf{S}_{n,m}$ of $\mathbf{S}$ can be written as the inner product of line $m \cdot N_xN_y + n$ of $\mathbf{A}$ and $\mathrm{vec}\{\mathbf{Y}\}$. Define

$$\begin{aligned}
\mathbf{c}(i,j) &= [c_0(i,j) \;\; \cdots \;\; c_{M_x-1}(i,j)]^T \\
&= \left[\mathrm{v}_{x_0}^{i*}\mathrm{v}_{x_0}^{j} \;\; \cdots \;\; \mathrm{v}_{x_{M_x-1}}^{i*}\mathrm{v}_{x_{M_x-1}}^{j}\right]^T \\
\mathbf{d}(k,l) &= \left[d_0(k,l) \;\; \cdots \;\; d_{M_y-1}(k,l)\right]^T \\
&= \left[\mathrm{v}_{y_0}^{k*}\mathrm{v}_{y_0}^{l} \;\; \cdots \;\; \mathrm{v}_{y_{M_y-1}}^{k*}\mathrm{v}_{y_{M_y-1}}^{l}\right]^T.
\end{aligned}$$

Using (24), we have

$$\mathbf{S}_{n,m} = \left[\mathbf{c}^T(i,j) \otimes \mathbf{d}^T(k,l)\right] \mathrm{vec}\{\mathbf{Y}\} \tag{26}$$

$$= \mathbf{d}^T(k,l)\mathbf{Y}\mathbf{c}(i,j) \tag{27}$$

where $i = \lfloor m/N_y \rfloor$, $j = \lfloor n/N_y \rfloor$, $k = \mathrm{mod}(m,N_y)$, and $l = \mathrm{mod}(n,N_y)$. Also, (26) and (27) are equivalent because

$(\mathbf{A}^T \otimes \mathbf{B})\text{vec}\{\mathbf{C}\} = \text{vec}\{\mathbf{BCA}\}$ whenever $\mathbf{BCA}$ is defined [30].

Note that $\mathbf{c}(i,j)$ and $\mathbf{d}(k,l)$ completely model the response of the $(n,m)$ pair of sensors, for all directions of arrival. This is already a more compact representation than before, since this model uses the separability of the array. All that is left is how to efficiently compute all of the responses for all pairs.

For $0 \le i,j < N_x$ and $0 \le k,l < N_y$, define

$$(i,j) \diamond (k,l) = \mathbf{d}^T(k,l)\mathbf{Y}\mathbf{c}(i,j) \tag{28}$$

and

$$\mathbf{T}_{j,i} = \begin{bmatrix} (i,j)\diamond(0,0) & \cdots & (i,j)\diamond(N_y-1,0) \\ (i,j)\diamond(0,1) & \cdots & (i,j)\diamond(N_y-1,1) \\ \vdots & & \vdots \\ (i,j)\diamond(0,N_y-1) & \cdots & (i,j)\diamond(N_y-1,N_y-1) \end{bmatrix}. \tag{29}$$

From the results above, it is easy to show that

$$\mathbf{S} = \begin{bmatrix} \mathbf{T}_{0,0} & \mathbf{T}_{0,1} & \cdots & \mathbf{T}_{0,N_x-1} \\ \mathbf{T}_{1,0} & \mathbf{T}_{1,1} & \cdots & \mathbf{T}_{1,N_x-1} \\ \vdots & \vdots & & \vdots \\ \mathbf{T}_{N_x-1,0} & \mathbf{T}_{N_x-1,1} & \cdots & \mathbf{T}_{N_x-1,N_x-1} \end{bmatrix}. \tag{30}$$

Even though one could determine $(i,j)\diamond(k,l)$ for $0 \le i,j < N_x$ and $0 \le k,l < N_y$ by directly evaluating (28), one should organize the computations to eliminate redundancy. Also, since in modern computer architectures the arithmetic units can process data faster than the main memory can provide via random accesses, one should maximize the locality of reference to ensure that the arithmetic operands are typically in the cache. In particular, the algorithm should promote sequential memory accesses so that the arithmetic units do not stall while waiting for a memory read. We will present this implementation below.

Let

$$\mathbf{t}_{i,j} = \text{vec}\{\mathbf{T}_{i,j}\} \tag{31}$$

$$\mathbf{Z} = [\mathbf{t}_{0,0} \quad \mathbf{t}_{1,0} \quad \ldots \quad \mathbf{t}_{N_x-1,N_x-1}]. \tag{32}$$

Given $\mathbf{Z}$, it is very easy to obtain $\mathbf{S}$, since every block $\mathbf{T}_{i,j}$ of $\mathbf{S}$ can be obtained by unstacking $\mathbf{t}_{i,j}$.

Define (33) and (34), shown in the equation at the bottom of the page. In comparison with (28), one can verify that

$$\mathbf{Z} = \mathbf{V_y}\mathbf{Y}\mathbf{V_x}^T. \tag{35}$$

Define $\Xi$ so that $\text{vec}\{\mathbf{S}\} = \Xi\text{vec}\{\mathbf{Z}\}$. (Note that $\Xi$ is a permutation.) Thus, $\text{vec}\{\mathbf{S}\} = \Xi(\mathbf{V_x} \otimes \mathbf{V_y})\text{vec}\{\mathbf{Y}\}$ and

$$\mathbf{A} = \Xi(\mathbf{V_x} \otimes \mathbf{V_y}). \tag{36}$$

Since $\Xi$ is a computationally efficient permutation and $(\mathbf{V_x} \otimes \mathbf{V_y})\text{vec}\{\mathbf{Y}\} = \text{vec}\{\mathbf{V_y}\mathbf{Y}\mathbf{V_x}^T\}$, (36) can be implemented as a fast transform (as follows).

From (30), it can be seen that each $\mathbf{T}_{i,j}$ contains the cross-covariance between two $N_y \times 1$ columns of sensors. Thus, $\mathbf{Z}$ is a reorganization of $\mathbf{S}$ which stacks these cross-covariances with a regularity that matches the row order of $\mathbf{V_x} \otimes \mathbf{V_y}$ (since $\text{vec}\{\mathbf{Z}\} = (\mathbf{V_x} \otimes \mathbf{V_y})\text{vec}\{\mathbf{Y}\}$).

We now make some remarks regarding computational cost. The direct product $\mathbf{Ay}$ in (16) requires approximately $(1/2)M_xM_yN_x^2N_y^2$ complex MACs when considering the Hermitian symmetry $\mathbf{S}$. Evaluating $(\mathbf{V_y}\mathbf{Y})\mathbf{V_x}^T$ and $\mathbf{V_y}(\mathbf{Y}\mathbf{V_x}^T)$ requires $N_y^2M_xM_y + N_x^2N_y^2M_x$ and $N_x^2M_xM_y + N_x^2N_y^2M_y$ complex MACs, respectively. Since $\mathbf{Y}$ is real-valued, the first product can be optimized and the costs drop to $(1/2)N_y^2M_xM_y + N_x^2N_y^2M_x$ and $(1/2)N_x^2M_xM_y + N_x^2N_y^2M_y$ complex MACs, respectively. Using the first expression and neglecting the time to obtain $\mathbf{S}$ from $\mathbf{Z}$, the relative speedup in terms of MACs is given by

$$\frac{\frac{1}{2}M_xM_yN_x^2N_y^2}{\frac{1}{2}N_y^2M_xM_y + N_x^2N_y^2M_x} = \frac{M_yN_x^2}{M_y + 2N_x^2}.$$

If the array geometry is symmetric with respect to the $y$ axis, then $\mathbf{V}_x$ has conjugate symmetry with respect to its middle row. An analogous statement applies to $\mathbf{V}_y$. If applicable, these symmetries can be used to further reduce the computational cost.

Recall that we introduced $\mathbf{Y}$ as having scan lines which realize an arbitrary Cartesian sampling of U-space. If $\{u_{x_i}\}$ and $\{u_{y_i}\}$ uniformly sample U-space, then $\mathbf{V_x}$ and $\mathbf{V_y}$ can be interpreted as DFT matrices for nonuniform frequency sampling. (This fact can be verified by explicitly writing $\mathbf{V_x}$ and $\mathbf{V_y}$ in terms of complex exponentials.) Therefore, for sufficiently large values of $N_x$ and $N_y$, a further optimization consists of using a

$$\mathbf{V_x} = \begin{bmatrix} c_0(0,0) & \cdots & c_{M_x-1}(0,0) \\ c_0(0,1) & \cdots & c_{M_x-1}(0,1) \\ \vdots & & \vdots \\ c_0(N_x-1,N_x-1) & \cdots & c_{M_x-1}(N_x-1,N_x-1) \end{bmatrix} \tag{33}$$

$$\mathbf{V_y} = \begin{bmatrix} d_0(0,0) & \cdots & d_{M_y-1}(0,0) \\ d_0(0,1) & \cdots & d_{M_y-1}(0,1) \\ \vdots & & \vdots \\ d_0(N_y-1,N_y-1) & \cdots & d_{M_y-1}(N_y-1,N_y-1) \end{bmatrix}. \tag{34}$$

fast nonequispaced Fourier transform (NFFT) [28] instead of each matrix product in (35). A rule of thumb obtained from numerical experiments is to use the NFFT for $N_x > 8$ or $N_y > 8$ and $M_x > 2^8$ or $M_y > 2^8$. Details regarding the performance with and without the NFFT are presented in Section V.

### B. Fast Adjoint Transform

As we present [21, Sec. II], with a measured spectral matrix $\mathbf{S}$, many computationally efficient image reconstruction methods require only fast implementations of $\mathbf{A}$ and $\mathbf{A}^H$ to estimate a source distribution $\hat{\mathbf{Y}}$. A computationally efficient reconstruction algorithm must have fast implementations of both; otherwise, the slow transform becomes the bottleneck for the solver.

Let $\bar{\mathbf{S}} \in \mathbb{C}^{N\times N}$ and $\bar{\mathbf{Y}} \in \mathbb{R}^{M_y\times M_x}$ so that $\mathrm{vec}\{\bar{\mathbf{Y}}\} = \mathbf{A}^H\mathrm{vec}\{\bar{\mathbf{S}}\}$. It follows from (36) that:

$$\mathbf{A}^H = \left(\mathbf{V}_\mathbf{x}^H \otimes \mathbf{V}_\mathbf{y}^H\right)\boldsymbol{\Xi}^H. \tag{37}$$

Since $\boldsymbol{\Xi}$ is a permutation, $\boldsymbol{\Xi}^{-1} = \boldsymbol{\Xi}^T = \boldsymbol{\Xi}^H$. If $\mathrm{vec}\{\bar{\mathbf{Z}}\} = \boldsymbol{\Xi}^T\mathrm{vec}\{\bar{\mathbf{S}}\}$, then

$$\bar{\mathbf{Y}} = \mathbf{V}_\mathbf{y}^H\bar{\mathbf{Z}}\mathbf{V}_\mathbf{x}^* \tag{38}$$

which is the fast implementation of $\mathbf{A}^H$. (Note that it has the same computational cost as the direct transform.)

If the U-space sampling is symmetric with respect to the $y$ axis, then $\mathbf{V}_\mathbf{x}$ has conjugate symmetry with respect to its center column. An analogous statement applies to $\mathbf{V}_\mathbf{y}$. If applicable, this symmetry can be used to further reduce the computational cost.

For separable arrays which are uniformly sampled in U-space, multiplication by $\mathbf{V}_\mathbf{x}$ and $\mathbf{V}_\mathbf{y}$ can again be optimized by using NFFTs, under the same considerations presented for the direct transform.

### C. Fast Direct-Adjoint Transform

Given the direct transform $\mathbf{A}$ and its adjoint $\mathbf{A}^H$, consider the transform given by $\mathbf{A}^H\mathbf{A}$. This composition will be used in [21, Sec. II] for image reconstruction, and in this section, we present a method of accelerating it further. Since $\boldsymbol{\Xi}^H = \boldsymbol{\Xi}^{-1}$, it follows from the previous results that $\mathrm{vec}\{\bar{\mathbf{Y}}\} = \mathbf{A}^H\mathbf{A}\mathrm{vec}\{\mathbf{Y}\}$ can be implemented as

$$\bar{\mathbf{Y}} = \mathbf{V}_\mathbf{y}^H\mathbf{V}_\mathbf{y}\mathbf{Y}\mathbf{V}_\mathbf{x}^T\mathbf{V}_\mathbf{x}^*. \tag{39}$$

This implementation is especially interesting when $N_x$ and $N_y$ are sufficiently large in comparison to $M_x$ and $M_y$, because it can be evaluated as

$$\bar{\mathbf{Y}} = \left(\mathbf{V}_\mathbf{y}^H\mathbf{V}_\mathbf{y}\right)\mathbf{Y}\left(\mathbf{V}_\mathbf{x}^T\mathbf{V}_\mathbf{x}^*\right) \tag{40}$$

with precomputed versions of $\mathbf{V}_\mathbf{y}^H\mathbf{V}_\mathbf{y}$ and $\mathbf{V}_\mathbf{x}^T\mathbf{V}_\mathbf{x}^*$, which are real valued.

Implementing the direct-adjoint transform with (39) can be much faster than using a composition of the direct and adjoint KAT, because for large problems, one can precompute $\mathbf{V}_\mathbf{y}^H\mathbf{V}_\mathbf{y}$ and $\mathbf{V}_\mathbf{x}^T\mathbf{V}_\mathbf{x}^*$. Furthermore, (39) can be parallelized more effectively, since it avoids applying $\boldsymbol{\Xi}$.

TABLE II
APPLICABILITY OF THE FFT, NFFT, NNFFT, AND KAT

| Transform | Geometry | U-space sampling | Exact? |
|---|---|---|---|
| FFT | Uniform | Uniform | Yes |
| NFFT | Arbitrary | Uniform | No |
| NNFFT | Arbitrary | Arbitrary | No |
| KAT | Separable | Separable | Yes |

The implementations, which use the NFFT for further acceleration, are at a disadvantage for the direct-adjoint transform, since one cannot precompute the equivalent of $\mathbf{V}_\mathbf{y}^H\mathbf{V}_\mathbf{y}$ and $\mathbf{V}_\mathbf{x}^T\mathbf{V}_\mathbf{x}^*$. Therefore, one is forced to use a composition of the previously presented transforms.

## IV. CONNECTIONS

In this section, we briefly describe how the KAT relates to the 2-D FFT, NFFT, and NNFFT. To our knowledge, the NFFT and NNFFT have never been applied to acoustic imaging. With the exception of the NNFFT, each transform is only suitable for specific array geometries or U-space sampling patterns. Transforms, which make more restrictive assumptions about the array geometry and U-space pattern, can generally be more computationally efficient, so the choice of which transform to use depends on a series of tradeoffs, summarized in Table II.

Assume that the sampled wavefield is a zero-mean random process which is stationary in time and homogenous in space.[2] Consider a sensor array consisting of $N$ microphones at coordinates $\mathbf{p}_0, \ldots, \mathbf{p}_{N-1} \in \mathbb{R}^3$. For a fixed frequency $\omega$, the cross-spectral matrix $\mathbf{S}(\omega) = \mathrm{E}\{\mathbf{x}(\omega)\mathbf{x}^H(\omega)\}$ is, by definition, a covariance matrix. For $0 \le m, n < N$, $[\mathbf{S}(\omega)]_{m,n}$ holds the frequency-domain cross-covariance of the wavefield between any two points whose coordinates differ by $\mathbf{p}_m - \mathbf{p}_n$. Let $P(\omega, \mathbf{k}) = |Y(\omega, \mathbf{k})|^2$ be the power spectral density when parameterized as a function of the wave number $\mathbf{k} = -(\omega/c)\mathbf{u} \in \mathbb{R}^3$, and $S(\omega, \Delta\mathbf{p})$ be the spectral covariance between two points whose coordinates differ by $\Delta\mathbf{p}$. It can be shown [29] that

$$P(\omega, \mathbf{k}) = \int_{\mathbb{R}^3} S(\omega, \Delta\mathbf{p})e^{+j\mathbf{k}^T\Delta\mathbf{p}}d\Delta\mathbf{p} \tag{41}$$

$$S(\omega, \Delta\mathbf{p}) = \frac{1}{(2\pi)^3}\int_{\mathbb{R}^3} P(\omega, \mathbf{k})e^{-j\mathbf{k}^T\Delta\mathbf{p}}d\mathbf{k} \tag{42}$$

which is essentially a generalization of the relationship between the cross-covariance and cross-spectral density for wide-sense stationary spatial-temporal processes, and is expressed as a Fourier transform.

Therefore, the knowledge of $S(\omega, \Delta\mathbf{p})$, limited to a finite set of baselines $\Delta\mathbf{p}$, allows us to approximate a discrete space version of $P(\omega, \mathbf{k})$, which is the image of interest. The following connections arise naturally from different ways of sampling these relations in order to evaluate them numerically for discrete space and discrete U-space.

[2]The random process model follows naturally from the fact that we do not know *a priori* what the source waveforms are. We model this process using its second-order statistics given by the cross-spectral matrix. Stationarity in time and homogeneity in space let us estimate the power spectral density of the process as a function of $\omega$ and $\mathbf{k}$, which is the acoustic image of interest.

### A. NFFT Imaging

A d-dimensional NDFT [28] (nonequispaced discrete Fourier transform) is defined by a set of arbitrary spatial nodes $\mathcal{X}$ and a frequency bandwidth vector $\mathbf{M} \in \mathbb{N}^d$. Each node $x_j$ belongs to the sampling set $\mathcal{X} = \{x_i \in [-(1/2),(1/2))^d : 0 \le i < N\}$ so that $|\mathcal{X}| = N$, where $|\cdot|$ indicates set cardinality. The index set

$$I_{\mathbf{N}} = \mathbb{Z}^d \cap \prod_{t=0}^{d-1} \left[-\frac{M_t}{2}, \frac{M_t}{2}\right) \tag{43}$$

defines a rectangular grid over which a function of interest is sampled.

Given as input a set of samples $h_{\mathbf{k}} \in \mathbb{C}$ for $\mathbf{k} \in I_{\mathbf{N}}$, the NDFT is defined as

$$\hat{h}_i = \sum_{\mathbf{k} \in I_{\mathbf{N}}} h_{\mathbf{k}} e^{-j2\pi \mathbf{k}^T \mathbf{x}_i} \tag{44}$$

for $0 \le i < N$.

The NFFT is a fast implementation of the NDFT obtained by interpolating an oversampled FFT. It is an approximate method which provides a very good compromise between accuracy and computational complexity.

Let $\mathbf{Y} \in \mathbb{C}^{M_y \times M_x}$ be an image obtained by uniform rectangular sampling of U-space with even $M_x$ and $M_y$, and sampling coordinates drawn from

$$\tilde{\mathrm{U}} = \left\{\frac{2i}{M_x}\right\}_{i=-M_x/2}^{M_x/2-1} \times \left\{\frac{2j}{M_y}\right\}_{j=-M_y/2}^{M_y/2-1}. \tag{45}$$

We now show that $\mathbf{S}$ can be obtained from $\mathbf{Y}$ by using a 2-D NFFT. Due to the linearity of the NFFT, it suffices to show that this transform is exact for an image containing one arbitrary unit impulse at coordinates $\mathbf{u}_0$, which must be in the U-space sampling grid.

Using (9), for arbitrary $-(M_x/2) \le m_0 < (M_x/2)$ and $-(M_y/2) \le n_0 < (M_y/2)$, define

$$\mathbf{u}_0 = \begin{bmatrix} \frac{2m_0}{M_x} \\ \frac{2n_0}{M_y} \\ \sqrt{1 - \left(\frac{2m_0}{M_x}\right)^2 - \left(\frac{2n_0}{M_y}\right)^2} \end{bmatrix} \tag{46}$$

and

$$\mathbf{Y}_{n,m} = \begin{cases} 1, & \text{if } (m,n) = \left(m_0 + \frac{M_x}{2}, n_0 + \frac{M_y}{2}\right) \\ 0, & \text{otherwise.} \end{cases} \tag{47}$$

By definition

$$\mathbf{S}_{r,s} = \left[\mathbf{v}(\mathbf{u}_0)\mathbf{v}^H(\mathbf{u}_0)\right]_{r,s} \tag{48}$$

$$= e^{j\frac{\omega}{c}\mathbf{u}_0^T(\mathbf{p}_r - \mathbf{p}_s)} \tag{49}$$

$$= e^{j2\pi \mathbf{u}_0^T \frac{(\mathbf{p}_r - \mathbf{p}_s)}{\lambda}} \tag{50}$$

$$= e^{j2\pi \frac{\mathbf{u}_0^T}{2} \frac{2(\mathbf{p}_r - \mathbf{p}_s)}{\lambda}}. \tag{51}$$

To obtain (51) by using the NFFT, rectangular U-space sampling and an arbitrary geometry of $N$ microphones, we use

$$\mathbf{M} = [M_x \quad M_y]^T \tag{52}$$

$$I_{\mathbf{N}} = \mathbb{Z}^2 \cap \left[-\frac{M_x}{2}, \frac{M_x}{2}\right) \times \left[-\frac{M_y}{2}, \frac{M_y}{2}\right) \tag{53}$$

$$\mathcal{X} = \left\{\mathbf{x}_i = \frac{2}{\lambda}\left(\mathbf{p}_{\lfloor i/N \rfloor} - \mathbf{p}_{\mathrm{mod}(i,N)}\right) \odot \begin{bmatrix} M_x^{-1} \\ M_y^{-1} \end{bmatrix} : 0 \le i < N^2\right\} \tag{54}$$

where $\odot$ represents the pointwise (Hadamard) product, and the baselines $\mathbf{p}_r - \mathbf{p}_s$ are represented only by their $x$ and $y$ coordinates. We now show that this parameterization of the NFFT produces the direct transform.

Once again

$$\mathbf{S}_{r,s} = e^{j2\pi \frac{\mathbf{u}_0^T}{2} \frac{2(\mathbf{p}_r - \mathbf{p}_s)}{\lambda}} = e^{j2\pi \left(\mathbf{u}_0^T \odot \left[\frac{M_x}{2} \ \frac{M_y}{2}\right]\right)\left(\frac{2}{\lambda}(\mathbf{p}_r - \mathbf{p}_s) \odot \left[M_x^{-1} \ M_y^{-1}\right]^T\right)} \tag{55}$$

where $\mathbf{u}_0$ is also represented only by its $x$ and $y$ components.

Comparing (55) with (44), the first term in parentheses clearly belongs to $I_{\mathbf{N}}$. Since for $0 \le i < N^2$, $(\mathbf{p}_{\lfloor i/N \rfloor} - \mathbf{p}_{\mathrm{mod}(i,N)})$ spans all possible baselines, the second term in parentheses belongs to $\mathcal{X}$. The enumeration given by $(\lfloor i/N \rfloor, \mathrm{mod}(i,N))$ indexes the elements of $\mathbf{S}$ row by row. Given the Hermitian symmetry of $\mathbf{S}$, this is equivalent to conjugating (55) and indexing the elements of $\mathbf{S}$ column by column (in the order of $\mathrm{vec}\{\cdot\}$), making (55) equivalent to (44).

It is possible to show that for uniform rectangular arrays with horizontal and vertical interelement spacings $d_x = d_y = \lambda/2$ (where $\lambda$ is the wavelength of the signal of interest), this NFFT reduces to a 2-D FFT. This implementation is not convenient for aeroacoustic imaging, since: 1) the constraint $d_x = d_y = \lambda/2$ can only be satisfied for one frequency, and we are interested in wideband operation; 2) the 2-D FFT is inefficient, since it ignores that image pixels significantly outnumber array sensors, and determines covariances for sensors that do not exist; 3) the 2-D FFT requires uniform rectangular geometries, which have their upper operating frequency constrained by the Nyquist–Shannon sampling theorem.

The NFFT has the advantage of allowing arbitrary array geometries, but as we will see in Section V, it is one order of magnitude slower than the KAT. Furthermore, as will be shown in Section VI, the KAT can be generalized to approximate the spherical wavefronts due to near-field sources. In contrast, the FFT, NFFT, and NFFT require a far-field assumption.

Finally, the KAT has the advantage of allowing separable (as opposed to uniform) U-space sampling grids. Acoustic images are often formed by clusters of distributed sources (for example, located over a model in a wind tunnel) and large regions with no significant sources. Thus, the KAT allows one to oversample the regions which are expected to have sources and undersample quiet regions, while maintaining low computational requirements.

### B. NNFFT Imaging

By dropping the uniform sampling constraint (45), one obtains the far-field array transform in its fullest generality. This transform can be accelerated with the nonequispaced in time and frequency fast Fourier transform (NNFFT). While the NNFFT is significantly slower than the KAT and the NFFT, it requires much less memory than the matrix representation of $\mathbf{A}$, which makes it useful for smaller problems that can be solved offline.

Given as input a set of samples $h_l \in \mathbb{C}$ for $0 \leq l < L$, the NNDFT is defined as

$$\hat{h}_i = \sum_{l=0}^{L-1} h_l e^{-j2\pi(\mathbf{v}_l \odot \mathbf{M})^T \mathbf{x}_i} \tag{56}$$

for $0 \leq i < N$, and arbitrary $\mathbf{v_l}, \mathbf{x}_i \in [-(1/2), (1/2))^d$. The NNFFT is a fast approximation of the NNDFT.

To obtain (51) using the NNFFT, arbitrary U-space sampling and an arbitrary geometry of $N$ microphones, we use $x_j \in \mathcal{X}$ and $v_l \in \mathcal{V}$, with

$$\mathbf{M} = [M_x \;\; M_y]^T \tag{57}$$

$$\mathcal{V} = \left\{ v_l = \left(\frac{u_{x_l}}{2}, \frac{u_{y_l}}{2}\right) \in \left[-\frac{1}{2}, \frac{1}{2}\right)^2 : 0 \leq l < M \right\} \tag{58}$$

$$\mathcal{X} = \left\{ \mathbf{x}_i = \frac{2}{\lambda} \left(\mathbf{p}_{\lfloor i/N \rfloor} - \mathbf{p}_{\mathrm{mod}(i,N)}\right) \odot \begin{bmatrix} M_x^{-1} \\ M_y^{-1} \end{bmatrix} : 0 \leq i < N^2 \right\} \tag{59}$$

which has the same form as (52)–(54), but allows arbitrary U-space sampling.

## V. Computational Cost

### A. Asymptotic Complexity

To simplify the following formulas, we will assume that $N_x = N_y$ and $M_x = M_y$. We will present the asymptotic complexity for the direct and adjoint transforms.

The product $\mathbf{Ay}$ in (16) requires approximately $(1/2)MN^2$ complex MACs to compute and, thus, has complexity $O(MN^2)$. For the KAT, the cost of computing $\boldsymbol{\Xi}$ can be neglected. Evaluating $(\mathbf{V_y Y})\mathbf{V}_\mathbf{x}^T$ with matrix multiplication requires $(1/2)MN + M^{1/2}N^2$ complex MACs and, thus, has complexity $O(MN + M^{1/2}N^2)$.

One can also evaluate $(\mathbf{V_y Y})\mathbf{V}_\mathbf{x}^T$ by interpreting each matrix product as a 1-D NFFT. Evaluating $\mathbf{V_y Y}$ requires $M_x$ 1-D NFFTs, each with cost $O(M_y \log M_y + N_y^2)$ [28]. The second product requires $N_y^2$ 1-D NFFTs, each with cost $O(M_x \log M_x + N_x^2)$. Assuming that $M^{1/2} > N$, the total complexity becomes $O(M \log M + M^{1/2}N)$. The direct NFFT and NNFFT implementations have complexity $O(M \log M + N^2)$ [28].

Table III summarizes these results. Note that since the asymptotic complexity is similar for most of the fast transforms, it simply guarantees that these methods will scale about as well as an FFT. Nevertheless, the constants hidden in the $O(\cdot)$ notation are significant. As we show next, the direct NNFFT implementation is much slower than a KAT with 1-D NFFTs, despite having similar asymptotic complexity. Also, since the $\mathbf{A}$ matrix is very large, memory bandwidth becomes the limiting factor for the explicit matrix representation. Thus, the constant hiding in the $O(\cdot)$ notation for the explicit matrix representation is greater than the MAC count suggests. Furthermore, for practical problem sizes, one does not have enough memory to store a full matrix representation and is forced to recompute the rows of $\mathbf{A}$ every time a matrix-vector product is required. This can dramatically increase the computational cost of the explicit matrix representation.

TABLE III
Asymptotic Complexity of the KAT, FFT, NFFT, NNFFT, and Explicit Matrix Representation

| Transform | Complexity |
|---|---|
| KAT with matrix multiplication | $O\left(MN + M^{1/2}N^2\right)$ |
| KAT with 1-D NFFTs | $O\left(M \log M + M^{1/2}N\right)$ |
| 2-D FFT | $O\left(M \log M\right)$ |
| 2-D NFFT | $O\left(M \log M + N^2\right)$ |
| 2-D NNFFT | $O\left(M \log M + N^2\right)$ |
| Explicit matrix representation | $O\left(MN^2\right)$ |

### B. Numerical Benchmarks

This section presents experiments to assess the execution times for the KAT, the NFFT, and the NNFFT. Even though the relative performance of algorithms based on matrix multiplication can be easily estimated in terms of MACs, actual runtimes can deviate significantly from these estimates for certain problem sizes. Indeed, for modern architectures, performance is strongly dependent on the interaction of parallel arithmetic units, memory bandwidth, cache size, and branch prediction, so that the number of floating-point operations only serves as an approximate measure of computational complexity.

The runtimes presented in Figs. 2 and 3 are averages collected over 10 s for each algorithm and problem size. All simulations were run on an Intel Core 2 Duo T9400 processor in 64-bit mode, using only one core. The permutation $\boldsymbol{\Xi}$, which obtains $\mathbf{S}$ from $\mathbf{Z}$, was written in ANSI C, the NFFT library was compiled with default optimizations as used by its authors, and all other functions were written in M-code for MATLAB R2008b. Since the code does not feature time-consuming loops, and MATLAB uses the Intel Math Kernel Library for matrix and vector arithmetic, the proposed transforms run very much like machine-specific tuned code. MATLAB and the NFFT use FFTW [31] for computing FFTs, so that they also run like machine-specific code. Thus, having the code written in MATLAB actually incurs negligible computational overhead when compared to an optimized implementation in C and machine-specific assembly code.

The computational cost of efficient convex optimization methods (and, in particular, of the regularized least-squares methods presented in [21, Sec. II] is completely dependent on the cost of applying $\mathbf{A}$, $\mathbf{A}^H$, $\boldsymbol{\Psi}$, and $\boldsymbol{\Psi}^H$. Since $\boldsymbol{\Psi}$ depends on the regularization method and can be chosen to be very fast, the bottleneck is on applying $\mathbf{A}$, $\mathbf{A}^H$, and possibly $\mathbf{A}^H\mathbf{A}$. Thus, from the runtimes of $\mathbf{A}$, $\mathbf{A}^H$, and $\mathbf{A}^H\mathbf{A}$ presented in this section, one can assume with good approximation that a $K$-fold

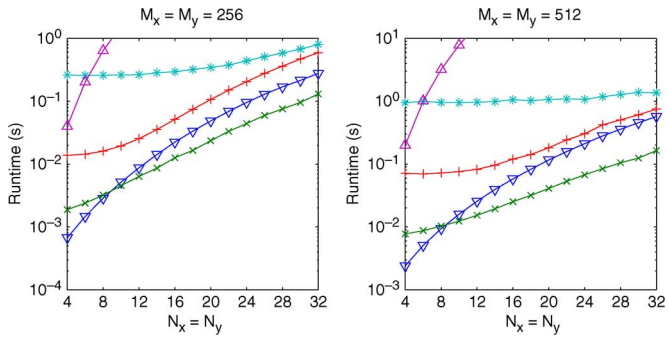


Fig. 2. Runtimes for the direct transform. $\triangledown$: KAT implemented with matrix multiplication, $\times$: KAT implemented with 1-D NFFTs replacing matrix multiplication, $+$: direct NFFT implementation with (52)–(54), $*$: direct NNFFT implementation with (57)–(59), $\triangle$: explicit matrix representation.

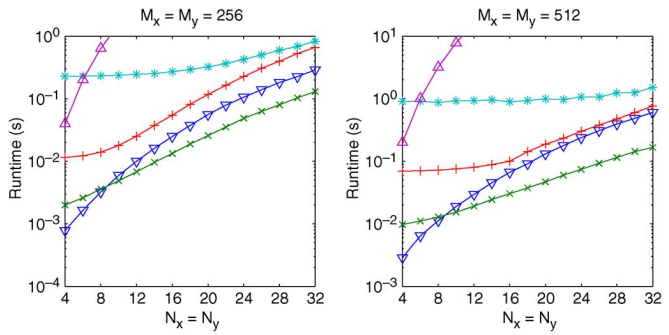


Fig. 3. Runtimes for the adjoint transform. $\triangledown$: KAT implemented with matrix multiplication, $\times$: KAT implemented with 1-D NFFTs replacing matrix multiplication, $+$: direct NFFT implementation with (52)–(54), $*$: direct NNFFT implementation with (57)–(59), $\triangle$: explicit matrix representation.

decrease in computational time translates to an algorithm which reconstructs an image $K$ times faster.

It is clear that the KAT with the NFFT optimization is the fastest transform for arrays with more than 64 elements ($N_x > 8$ and $N_y > 8$). This is the case because as $\mathbf{V_x}$ and $\mathbf{V_y}$ grow, the NFFT scales better than matrix multiplication. The direct implementation with the NFFT is useful if one must have an arbitrary array geometry, but it has the drawback of being around an order of magnitude slower and requiring a far-field approximation. (As we show in Section VI, the KAT can be extended for near-field imaging.) The direct implementation with the NNFFT is by far the slowest. For all implementations, the direct and adjoint transforms perform similarly.

Fig. 4 presents runtimes for the direct-adjoint composition. The implementation using (39) analyzes the problem size and automatically selects the optimal order for matrix multiplication. For large values of $N_x$ and $N_y$, it also uses precomputed versions of $\mathbf{V_y}^H\mathbf{V_y}$ and $\mathbf{V_x}^T\mathbf{V_x}^*$, which makes the computational complexity depend only on $M_x$ and $M_y$. The NFFT implementation uses a composition of the direct and adjoint transforms, without any additional optimizations.

As shown in [21], under a far-field assumption and uniform U-space sampling, the direct-adjoint composition reduces to a 2-D convolution of the input image with the array point-spread function. Thus, it can be accelerated with a 2-D FFT (with zero padding to prevent edge effects). Fig. 4 shows that the direct-adjoint composition implemented with the KAT also outperforms

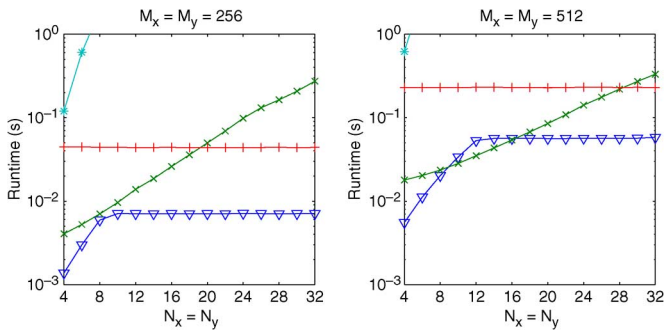


Fig. 4. Runtimes for the direct-adjoint composition. $\triangledown$: KAT implemented with (39), $\times$: KAT implemented with the composition $\mathbf{A}$ and $\mathbf{A}^H$, with the 1-D NFFT optimization, $+$: 2-D FFT-accelerated convolution, $*$: composition of the explicit matrix representation.

2-D FFT accelerated convolutions. As we show in the following section, the KAT can be generalized to near-field scenarios, allowing us to drop the far-field assumption. Note that for near-field cases, the direct-adjoint composition no longer reduces to a convolution, and KAT becomes the only fast transform suitable for imaging.

Finally, the KAT has the additional advantage of being easy to implement and parallelize, since it only requires relatively small matrix multiplications and computationally efficient permutations.

## VI. Near-Field Imaging

Up to this point, we have assumed that the sources were located in the far field. Thus, we used a plane-wave model. In this section, we show how to generalize the KAT and address near-field scenarios, where one has spherical wavefronts.

Note that the KAT does not impose any structure onto the array manifold vector other than its separability. The specific far-field representation was only chosen for convenience, since for any Cartesian geometry, the far-field array manifold vector is separable. Nothing prevents us from choosing a different separable representation that is more suitable for the near-field case. In this section, we show that the problem of finding the best separable representation can be recast as a rank-1 approximation of a rearranged version of $\mathbf{\Xi}^{-1}\mathbf{A} = \mathbf{\Xi}^T\mathbf{A}$. By using a rank-$K$ approximation (for $K > 1$), one is able to obtain an arbitrarily accurate model for near-field propagation, while maintaining low computational requirements.

To simplify the language in this section, we will use the following notation. Given suitably sized matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$, we use $\mathbf{C} = \underline{\mathbf{A}}(\mathbf{B})$ to denote $\mathrm{vec}\{\mathbf{C}\} = \mathbf{A}\mathrm{vec}\{\mathbf{B}\}$.

We approximate $\mathbf{A}$ by $\breve{\mathbf{A}} = \mathbf{\Xi}(\sum_{k=1}^{K} \mathbf{C}_k \otimes \mathbf{D}_k)$, for small values of $K$. Note that $\mathbf{S} = \breve{\mathbf{A}}(\mathbf{Y})$ can be efficiently implemented as $\mathbf{S} = \mathbf{\Xi}(\sum_{k=1}^{K} \mathbf{D}_k\mathbf{Y}\mathbf{C}_k^T)$. Compared with (36), we are approximating the near-field transform $\mathbf{A}$ with a series of $K$ separable transforms, to which we can apply the KAT.

Let us consider the problem of approximating a generic $\mathbf{B} \in \mathbb{C}^{m\times n}$ with $m = m_1m_2$ and $n = n_1n_2$ with a sum of Kronecker products, so that

$$\min_{\{\mathbf{C}_k\},\{\mathbf{D}_k\}} \left\| \mathbf{B} - \sum_{k=1}^{K} \mathbf{C}_k \otimes \mathbf{D}_k \right\|_F$$

where $\mathbf{C}_k \in \mathbb{C}^{m_1 \times n_1}$ and $\mathbf{D}_k \in \mathbb{C}^{m_2 \times n_2}$ for $1 \leq k \leq K$. This problem is addressed in [32], where it is shown to be equivalent to

$$\min_{\{\mathbf{C}_k\},\{\mathbf{D}_k\}} \left\| \mathcal{R}(\mathbf{B}) - \sum_{k=1}^{K} \mathrm{vec}\{\mathbf{C}_k\}\mathrm{vec}\{\mathbf{D}_k\}^T \right\|_F \tag{60}$$

where $\mathcal{R}(\cdot)$ is a matrix rearrangement operator so that $\mathcal{R}(\mathbf{B}) \in \mathbb{C}^{m_1 n_1 \times m_2 n_2}$. This is a low-rank approximation problem which can be solved with the SVD of $\mathcal{R}(\mathbf{B})$.

For our purposes, we approximate $\mathbf{B} = \mathbf{\Xi}^{-1}\mathbf{A}$. We note that $\mathbf{\Xi}$ is the key to a successful low-rank decomposition. As we show later in this section, using $\mathbf{B} = \mathbf{A}$ is not useful, since $\mathcal{R}(\mathbf{A})$ has too many significant singular values.

Computing the dominant singular values and vectors of $\mathcal{R}(\mathbf{B})$ is not trivial, since, in practice, $\mathcal{R}(\mathbf{B})$ is too large to be stored explicitly in memory. Nevertheless, one can use the Lanczos methods [32], [33] which only require the implementation of the matrix-vector products $\mathcal{R}(\mathbf{B})\boldsymbol{\alpha}$ and $\mathcal{R}(\mathbf{B})^H\boldsymbol{\beta}$ for arbitrary $\boldsymbol{\alpha}, \boldsymbol{\beta}$. One can also use approximate SVD methods which are designed to require a small number of passes over $\mathcal{R}(\mathbf{B})$ (e.g., [34] and [35]).

Using the definition of $\mathcal{R}(\cdot)$ from [32], it can be shown that

$$\mathcal{R}(\mathbf{B})^T = \begin{bmatrix} \mathbf{Z}_{0,0} & \cdots & \mathbf{Z}_{M_x-1,0} \\ \vdots & & \vdots \\ \mathbf{Z}_{0,M_y-1} & \cdots & \mathbf{Z}_{M_x-1,M_y-1} \end{bmatrix} \tag{61}$$

$$\mathbf{Z}_{m,n} = \underline{\underline{\Xi}}\left(\mathbf{v}\left(u_{x_m}, u_{y_n}\right)\mathbf{v}^H\left(u_{x_m}, u_{y_n}\right)\right). \tag{62}$$

Since $\mathbf{v}(u_{x_m}, u_{y_n})$ can be precomputed for $0 < m < M_x$ and $0 < n < M_y$ and $\mathbf{\Xi}$ is a very fast permutation, $\mathcal{R}(\mathbf{B})\boldsymbol{\alpha}$ and $\mathcal{R}(\mathbf{B})^H\boldsymbol{\beta}$ can be evaluated with relative efficiency. Indeed, using the Lanczos method from [33], $N = 64$ and $M_x = M_y = 256$, we can solve (60) for $K = 8$ in 8 min on an Intel Core 2 Duo 2.4-GHz processor, using only one core. Note that this procedure only has to be performed once.

The decomposition obtained with (60) is especially useful for $K > 1$. Indeed, even in the presence of strong near field effects, $\mathcal{R}(\mathbf{B})$ can be well approximated by a low-rank decomposition. Even though the transform cost grows linearly with $K$, due to the Kronecker representation, the cost of applying each $\mathbf{C}_k \otimes \mathbf{D}_k$ is very small, so a transform with $K = 8$ is still very fast.

Since we discretize the focal surface with Cartesian sampling grids, some images may have invalid regions. For example, one can parameterize a spherical half-shell with radius $r_0$ by using $u_x = r_0 \sin\phi\cos\theta$ and $u_y = r_0 \sin\phi\sin\theta$ for $\|\mathbf{u}\| \leq r_0$, so that source locations for $\|\mathbf{u}\| > r_0$ are invalid. Note that this implies that some $\mathbf{Z}_{m,n}$ in $\mathcal{R}(\mathbf{B})$ are not defined. Referring to (60), the optimal $\mathbf{C}_k, \mathbf{D}_k$ are now given by

$$\min_{\{\mathbf{C}_k\},\{\mathbf{D}_k\}} \left\| \mathbf{W} \odot \left( \mathcal{R}(\mathbf{B}) - \sum_{k=1}^{K} \mathrm{vec}\{\mathbf{C}_k\}\mathrm{vec}\{\mathbf{D}_k\}^T \right) \right\|_F$$

where $\odot$ is the pointwise (Hadamard) product and $\mathbf{W}$ is a binary mask set to 1 for valid elements and 0 for invalid elements of $\mathcal{R}(\mathbf{B})$. This masked-SVD problem was considered in [36], and can be solved by iterating

$$\mathbf{E}^{(i)} = \mathrm{LRA}_{\mathrm{K}}\left(\mathbf{W} \odot \mathcal{R}(\mathbf{B}) + (\mathbb{1} - \mathbf{W}) \odot \mathbf{E}^{(i-1)}\right)$$

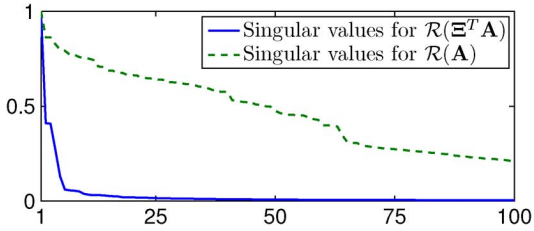


Fig. 5. First 100 singular values for $\mathcal{R}(\mathbf{A})$ and $\mathcal{R}(\mathbf{\Xi}^T\mathbf{A})$, out of a total of 16 384, normalized to 1.

where $\mathrm{LRA}_{\mathrm{K}}(\cdot)$ is a rank-$K$ approximation as computed by the SVD, $\mathbf{E}^{(i)} = \sum_{k=1}^{K} \mathrm{vec}\{\mathbf{C}_k^{(i)}\}\mathrm{vec}\{\mathbf{D}_k^{(i)}\}^T$ and $\mathbf{E}^{(0)} = \mathbf{0}$. In our experiments, 1 or 2 iterations have shown to be sufficient for a good fit.

Fig. 5 compares the first 100 (out of a total of 16384) singular values for $\mathcal{R}(\mathbf{A})$ and $\mathcal{R}(\mathbf{\Xi}^T\mathbf{A})$. In this experiment, $\mathbf{A}$ models $256 \times 256$ sources radiating at 9 kHz, located over a spherical half-shell with a radius of 1.0 m, as measured by an $8 \times 8$ separable array with a $30 \times 30$-cm aperture. (This configuration is the same used in the examples from [21].) The sharp decay of the curve for $\mathcal{R}(\mathbf{\Xi}^T\mathbf{A})$ highlights the importance of $\mathbf{\Xi}$ in enabling an accurate low-rank approximation. For lower frequencies, the singular values show an even sharper roll-off. Also, this transform models the complete hemisphere. By modeling a smaller field of view, near-field effects are not as severe, and the singular values also decay faster.

Note that by using a rank-$K$ approximation, we obtain a transform with a computational cost that is $K$ times larger than the far-field KAT presented in the previous sections. Nevertheless, as we show in [21], $K$ will be small enough that this penalty is not significant. In fact, we will show that it is possible to compensate for strong near-field effects with $K = 8$, which makes the KAT about as fast as the NFFT, while being able to model arbitrary near-field focal surfaces.

## VII. Conclusion

This paper presents the KAT, which was designed to enable computationally efficient and accurate acoustic imaging. To obtain fast implementation, we assumed a separable microphone array, source parameterization in U-space, and far-field sources. The KAT transforms back and forth between a hypothetical source distribution and the corresponding covariance matrix which would be measured by the array. Another contribution is the novel application of the NFFT and NNFFT to acoustic imaging, which we used as baselines against which our proposal was compared.

The KAT is orders of magnitude faster than equivalent implementations using explicit matrix representations. Despite the fact that the NFFT and NNFFT are already fast transforms, we have shown in Section V that our proposal is at least one order of magnitude faster than them. In contrast with FFT- and NFFT-based transforms, the KAT allows arbitrary Cartesian samplings of the source distributions, which let one oversample regions with sources and undersample silent areas without performance degradation. Also, in contrast with the NFFT, the KAT makes no numerical approximations, and can be more easily implemented and parallelized, since it only requires relatively small matrix products and simple permutations.

Even though the KAT was motivated with the far-field assumption, it does not impose any structure onto the array manifold vector other than its separability. We have used this fact to extend it for near-field imaging, providing a computationally efficient approximation of the exact near-field transform.

Future work involves developing fast transforms for other array geometries. In contrast with the KAT, which is exact for far-field sources, transforms for other geometries will most likely require approximations to obtain good performance. Furthermore, the KAT can be generalized for correlated distributions, which we also intend to address in future work.

## REFERENCES


[1] W. Home, K. James, T. Arledge, P. Sodermant, N. Burnside, and S. Jaeger, "Measurements of 26%-scale 777 airframe noise in the NASA Ames 40- by 80-foot wind tunnel," presented at the 11th AIAA/CEAS Aeroacoust. Conf., Monterey, CA, 2005.

[2] W. Humphreys and T. Brooks, "Noise spectra and directivity for scale-model landing gear," presented at the 13th AIAA/CEAS Aeroacoust. Conf., Rome, Italy, 2007.

[3] S. Lee, "Phased-array measurement of modern regional aircraft turbofan engine noise," presented at the 12th AIAA/CEAS Aeroacoust. Conf., Cambridge, MA, 2006.

[4] S. Oerlemans, P. Sijtsma, and B. Mendez Lopez, "Location and quantification of noise sources on a wind turbine," *J. Sound Vibr.*, vol. 299, no. 4–5, pp. 869–883, 2007.

[5] A. O'Donovan, R. Duraiswami, and D. Zotkin, "Imaging concert hall acoustics using visual and audio cameras," in *Proc. ICASSP*, 2008, pp. 5284–5287.

[6] M. Wax and T. Kailath, "Optimum localization of multiple sources by passive arrays," *IEEE Trans. Acoust., Speech Signal Process.*, vol. ASSP-31, no. 5, pp. 1210–1217, Oct. 1983.

[7] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. AP-34, no. 3, pp. 276–280, Mar. 1986.

[8] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 7, pp. 984–995, Jul. 1989.

[9] C. Zhang, D. Florencio, D. Ba, and Z. Zhang, "Maximum likelihood sound source localization and beamforming for directional microphone arrays in distributed meetings," *IEEE Trans. Multimedia*, vol. 10, no. 3, pp. 538–548, Apr. 2008.

[10] M. Brandstein and H. Silverman, "A robust method for speech signal time-delay estimation inreverberant rooms," in *Proc. ICASSP*, 1997, vol. 1, pp. 375–378.

[11] Z. Wang, J. Li, P. Stoica, T. Nishida, and M. Sheplak, "Constant-beamwidth and constant-powerwidth wideband robust Capon beamformers for acoustic imaging," *J. Acoust. Soc. Amer.*, vol. 116, p. 1621, 2004.

[12] S. Yan, Y. Ma, and C. Hou, "Optimal array pattern synthesis for broadband arrays," *J. Acoust. Soc. Amer.*, vol. 122, p. 2686, 2007.

[13] J. Li, Y. Xie, P. Stoica, X. Zheng, and J. Ward, "Beampattern synthesis via a matrix approach for signal power estimation," *IEEE Trans. Signal Process.*, vol. 55, no. 12, pp. 5643–5657, Dec. 2007.

[14] Y. Wang, J. Li, P. Stoica, M. Sheplak, and T. Nishida, "Wideband relax and wideband clean for aeroacoustic imaging," *J. Acoustic. Soc. Amer.*, vol. 115, p. 757, 2004.

[15] R. Dougherty and R. Stoker, "Sidelobe suppression for phased array aeroacoustic measurements," in *Proc. 4th AIAA/CEAS Aeroacoust.s Conf.*, 1998, pp. 235–245.

[16] R. Dougherty, "Extensions of DAMAS and benefits and limitations of deconvolution in beamforming," presented at the 11th AIAA/CEAS Aeroacoustics Conf., Monterey, CA, 2005.

[17] K. Ehrenfried and L. Koop, "Comparison of iterative deconvolution algorithms for the mapping of acoustic sources," *AIAA J.*, vol. 45, no. 7, p. 1584, 2007.

[18] E. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, no. 8, p. 1207, 2006.

[19] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[20] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[21] F. Ribeiro and V. Nascimento, "Fast transforms for acoustic imaging—Part II: Applications and extensions," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. XXX–XXX, Aug. 2011.

[22] L. He, T. Chang, S. Osher, T. Fang, and P. Speier, "MR image reconstruction by using the iterative refinement method and nonlinear inverse scale space methods," *UCLA CAM Rep.*, vol. 6, p. 35, 2006.

[23] M. Lustig, D. Donoho, and J. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Resonan. Med.*, vol. 58, no. 6, pp. 1182–1195, 2007.

[24] D. Malioutov, M. Cetin, and A. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 3010–3022, Aug. 2005.

[25] T. Yardibi, J. Li, P. Stoica, and L. Cattafesta, III, "Sparsity constrained deconvolution approaches for acoustic source mapping," *J. Acoust. Soc. Amer.*, vol. 123, p. 2631, 2008.

[26] B. Zimmermann and C. Studer, "FPGA-based real-time acoustic camera prototype," in *Proc. ISCAS*, 2010, p. 1419.

[27] X. Huang, "Real-time algorithm for acoustic imaging with a microphone array.," *J. Acoust. Soc. Amer.*, vol. 125, no. 5, 2009.

[28] J. Keiner, S. Kunis, and D. Potts, "Using NFFT 3—A software library for various nonequispaced fast fourier transforms," *ACM Trans. Math. Softw.*, vol. 36, no. 4, p. 19, 2009.

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

[30] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1990.

[31] M. Frigo and S. Johnson, "The design and implementation of FFTW 3," *Proc. IEEE*, vol. 93, no. 2, pp. 216–231, Feb. 2005.

[32] C. Van Loan and N. Pitsianis, "Approximation with Kronecker products," in *Linear Algebra for Large Scale and Real Time Applications*, M. Moonen and G. Golub, Eds. Norwell, MA: Kluwer, 1992, pp. 293–314.

[33] J. Baglama and L. Reichel, "Augmented implicitly restarted Lanczos bidiagonalization methods," *SIAM J. Scientif. Comput.*, vol. 27, no. 1, pp. 19–42, 2006.

[34] P. Drineas, R. Kannan, and M. Mahoney, "Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix," *SIAM J. Comput.*, vol. 36, no. 1, pp. 158–183, 2007.

[35] N. Halko, P. Martinsson, and J. Tropp, "Finding structure with randomness: stochastic algorithms for constructing approximate matrix decompositions," California Inst. Tech., Tech. rep. ACM 2009-05, Sep. 2009.

[36] N. Srebro and T. Jaakkola, "Weighted low-rank approximations," in *Proc. ICML*, 2003, pp. 720–727.


**Flávio P. Ribeiro** (S'09) received the B.S. degree in electrical engineering from Escola Politécnica, University of São Paulo, São Paulo, Brazil, in 2005, and the B.S. degree in mathematics from the Institute of Mathematics and Statistics, University of São Paulo, in 2008. He is currently pursuing the Ph.D. degree in electrical engineering from the Escola Politécnica, University of São Paulo.

From 2007 to 2009, he was a Hardware Engineer with Licht Labs, where he developed controllers for power transformers and substations. In the Summers of 2009 and 2010, he was a Research Intern with Microsoft Research Redmond. His research interests include array signal processing, multimedia signal processing, and computational linear algebra.

Mr. Ribeiro was a recipient of the Best Student Paper Award at ICME 2010.

**Vítor H. Nascimento** (M'01) was born in São Paulo, Brazil. He received the B.S. and M.S. degrees in electrical engineering from the University of São Paulo, in 1989 and 1992, respectively, and the Ph.D. degree from the University of California, Los Angeles, in 1999.

From 1990 to 1994, he was a Lecturer with the University of São Paulo, and in 1999 he joined the faculty at the same school, where he is now an Associate Professor. His research interests include signal-processing theory and applications, robust and nonlinear estimation, and applied linear algebra.

Prof. Nascimento was a recipient of the IEEE Signal Processing Society (SPS) Best Paper Award in 2002. He served as an Associate Editor for IEEE SIGNAL PROCESSING LETTERS from 2003 to 2005, for the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2005 to 2008, and for the *EURASIP Journal on Advances in Signal Processing* from 2006 to 2009. He is currently a member of the IEEE-SPS Signal Processing Theory and Methods Technical Committee.