A FAST TRANSFORM FOR ACOUSTIC IMAGING WITH SEPARABLE ARRAYS

Flávio P. Ribeiro, Vítor H. Nascimento

Electronic Systems Engineering Dept., Universidade de São Paulo
{fr,vitor}@lps.usp.br

ABSTRACT

Acoustic imaging is a computationally intensive and ill-conditioned inverse problem, which involves estimating high resolution source distributions with large microphone arrays. In this paper we show how to significantly decrease its computational cost with a fast transform designed for separable array geometries. This transform provides a natural and elegant way of accelerating beamforming, deconvolution methods and regularized least-squares solvers. We accelerate image deconvolution by 10x with respect to FFT-based methods, and accelerate other important imaging algorithms (based on explicit matrix multiplication) by even larger factors. Because of these gains in computational speed, one can reconstruct images with higher resolutions than previously possible, and also enable more accurate reconstruction techniques, opening new and exciting possibilities for acoustic imaging.

Index Terms— array processing, fast transform, acoustic imaging, array imaging, regularized least-squares, deconvolution.

1. INTRODUCTION

Acoustic imaging with microphone arrays has become a standard tool for studying aeroacoustic sources. It is routinely used to measure the noise generated by engines, turbines, vehicles and aircraft [1] for aerodynamic design and noise reduction purposes. Acoustic imaging has also been used to visualize the reverberant structure of concert halls [2].

The fastest method for aeroacoustic imaging uses beamforming, which produces the source distribution of interest convolved with the array point spread function (PSF). Since a typical microphone array has at most a few hundred elements, its PSF can be quite large, such that beamforming produces a very smeared image.

To partially undo this effect, several iterative deconvolution techniques have been proposed [3, 4]. More recently, regularized covariance fitting [5] has been shown to deliver more accurate results, but with very high computational costs.

For large imaging problems, even beamforming can be computationally intensive if implemented for a large generic array. Furthermore, in the absence of a fast transform, any iterative method which requires transforming back and forth between image iterates and the measured data can easily become computationally intractable. Iterative deconvolution methods for acoustic imaging such as DAMAS2 [4] only have acceptable performance because they operate over the beamformed image and the array PSF, without ever transforming back to the measured data.

This paper presents a fast transform developed for acoustic imaging, which can be used to accelerate beamforming, deconvolution and covariance fitting methods given the assumption of a separable planar array geometry. By allowing algorithms to transform back to the measured data, this transform enables the practical use of state-of-the-art general purpose regularized least-squares solvers, obviating sub-optimal methods which are only preferred for computational reasons.

Our proposal is an exact transform, which can be efficiently implemented using only matrix products and simple permutations. As we will show, PSF convolutions implemented with this transform outperform FFT-accelerated convolutions by a factor of 10. This transform outperforms explicit matrix multiplication by an even greater margin, which allows it to accelerate a wide range of algorithms for acoustic imaging.

This paper is organized as follows: Section 2 provides definitions and further motivates the need for fast transforms. Section 3 introduces the fast direct transform, its adjoint and the fast composition of the direct and adjoint. Section 4 has applications, and Section 5 has our conclusions.

2. PRELIMINARIES

In the following, we use the superscripts $\cdot^T$ to denote matrix or vector transpose, $\cdot^H$ to denote Hermitian transpose, and $\cdot^*$ to denote complex conjugate. The remainder of a/b, for $a, b \in \mathbb{Z}_+$ is written as $\mathrm{mod}\,(a, b)$. Round-off of $x \in \mathbb{R}$ towards $-\infty$ is denoted by $\lfloor x \rfloor$.

Consider a planar array of N microphones with coordinates $\mathbf{p}_0$, ..., $\mathbf{p}_{N-1} \in \mathbb{R}^3$. Suppose the wave field of interest can be modeled as generated by the superposition of M point sources with coordinates $\mathbf{q}_0, ..., \mathbf{q}_{M-1} \in \mathbb{R}^3$, where M may be large to obtain an accurate model. The $N \times 1$ array output for a frequency ω is modeled as

$$\mathbf{x}\,(\omega) = \mathbf{V}\,(\omega)\,\mathbf{f}\,(\omega) + \boldsymbol{\eta}\,(\omega)\,, \tag{1}$$

where $\mathbf{V}\,(\omega) = \left[\, \mathbf{v}\,(\mathbf{q}_0, \omega)\;\; \mathbf{v}\,(\mathbf{q}_1, \omega)\;\; \cdots\;\; \mathbf{v}\,(\mathbf{q}_{M-1}, \omega) \,\right]$ is the array manifold matrix, $\mathbf{f}\,(\omega) = \left[\, f_0\,(\omega)\;\; f_1\,(\omega)\;\; \cdots\;\; f_{M-1}\,(\omega) \,\right]^T$ is the frequency domain signal waveform and $\boldsymbol{\eta}\,(\omega)$ is uncorrelated noise.

Define $\mathbf{u}_m = \mathbf{q}_m / \left\| \mathbf{q}_m \right\|$, the look direction for source m. The far-field array manifold vector for source m is given by

$$\mathbf{v}\,(\mathbf{u}_m, \omega) = \left[\, \mathrm{e}^{j \frac{\omega}{c} \mathbf{u}_m^T \mathbf{p}_0}\;\; \ldots\;\; \mathrm{e}^{j \frac{\omega}{c} \mathbf{u}_m^T \mathbf{p}_{N-1}} \,\right]^T. \tag{2}$$

If θ and ϕ are the azimuth and elevation angles, one can parameterize the unit half-sphere by defining

$$u_x\,(\theta, \phi) = \sin\phi \cos\theta \qquad u_y\,(\theta, \phi) = \sin\phi \sin\theta,$$

such that

$$\mathbf{u} = \left[\, u_x\;\; u_y\;\; \sqrt{1 - u_x^2 - u_y^2} \,\right]^T \tag{3}$$

for $u_x^2 + u_y^2 \leq 1$. Recall that under a far-field approximation, uniform sampling in U-space (where $\mathrm{U} = [-1, 1]^2$) makes point-spread functions shift-invariant. Since the array is planar, it follows from (2) that we can ignore the z coordinate of $\mathbf{u}_m$ if the array is placed horizontally, so the z coordinates of $\mathbf{p}_0$, ..., $\mathbf{p}_{N-1}$ are zero. Thus, we will abuse notation and use $\mathbf{u}_m, \mathbf{p}_n \in \mathbb{R}^2$.

Let $\mathbf{S}_\mathbf{x}\,(\omega) = \mathrm{E}\left\{ \mathbf{x}\,(\omega)\, \mathbf{x}^H\,(\omega) \right\}$ be the array cross spectral matrix. If $\mathbf{x}_0\,(\omega)$, ..., $\mathbf{x}_{L-1}\,(\omega)$ correspond to L frequency domain

978-1-4577-0539-7/11/$26.00 ©2011 IEEE

snapshots, the spectral matrix can be estimated with

$$\mathbf{S}_{\mathbf{x}}(\omega) = \frac{1}{L}\sum_{l=0}^{L-1} \mathbf{x}_l(\omega)\,\mathbf{x}_l^H(\omega). \tag{4}$$

Processing $\mathbf{S}_{\mathbf{x}}(\omega)$ instead of each $\mathbf{x}_l(\omega)$ is typically more convenient, because $\mathbf{S}_{\mathbf{x}}(\omega)$ carries only the relative phase shifts between microphones and is the result of averaging, such that it has less noise content. Indeed, each $\mathbf{x}_l(\omega)$ has a phase shift which is equal for every element but unknown, which disappears when computing $\mathbf{S}_{\mathbf{x}}(\omega)$. We will assume narrow-band processing and omit the argument ω. To simplify the notation, $\mathbf{S}_{\mathbf{x}}(\omega)$ will be written as $\mathbf{S}$.

Assuming that the noise is spatially white and uncorrelated with the sources of interest, we have

$$\mathbf{S} = \mathbf{V}\,\mathrm{E}\left\{\mathbf{f}\mathbf{f}^H\right\}\mathbf{V}^H + \sigma^2\mathbf{I}, \tag{5}$$

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

One can map the M point sources to coordinates $\{\mathbf{u}_i\}_{i=0}^{M-1}$ located in a sufficiently fine grid in U-space. This representation produces a two-dimensional image, where pixel coordinates correspond to source locations, and pixel values correspond to source powers. Note that in (5), assuming that the sources are uncorrelated (a nearly universal assumption for acoustic imaging) implies that $\mathrm{E}\left\{\mathbf{f}\mathbf{f}^H\right\}$ is diagonal. Furthermore, the diagonal of $\mathrm{E}\left\{\mathbf{f}\mathbf{f}^H\right\}$ has the power radiated from each direction, i.e., is a vectorized version of the image.

Given a $M_x \times M_y$ acoustic image, define $M = M_xM_y$ and let $\{\mathbf{u}_m\}_{m=0}^{M-1}$ be an enumeration of all pixel coordinates in U-space. Let $\mathbf{v}(\mathbf{u}_m)$ be the array manifold vector when steered towards $\mathbf{u}_m$. For sources at $\{\mathbf{u}_m\}_{m=0}^{M-1}$ radiating with powers $\left\{|Y(\mathbf{u}_m)|^2\right\}_{m=0}^{M-1}$ and in the absence of noise,

$$\mathbf{S} = \sum_{m=0}^{M-1} |Y(\mathbf{u}_m)|^2\,\mathbf{v}(\mathbf{u}_m)\,\mathbf{v}^H(\mathbf{u}_m). \tag{6}$$

Many reconstruction algorithms iteratively compute an estimated image $\left\{\left|\hat{Y}(\mathbf{u}_m)\right|^2\right\}_{m=0}^{M-1}$ and compare the corresponding $\hat{\mathbf{S}}$ obtained through (6) with the measured values obtained from (4). Unless the image is very sparse, (6) becomes computationally intractable. For instance, for a 256 element array and a 256 × 256 acoustic image, (6) requires 2^{32} complex multiply-accumulate instructions to compute. This cost is excessive for a transform intended to be used in an iterative method. In the following section we describe how to obtain an efficient transform to compute (6).

3. FAST IMAGING TRANSFORMS

3.1. Fast direct transform

Define $\mathbf{y} = \left[\, |Y(\mathbf{u}_0)|^2 \ \cdots \ |Y(\mathbf{u}_{M-1})|^2 \,\right]^T$. Let us write (6) as a linear transform $\mathbf{A}$ such 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 $\mathrm{v}_{u_m}^i$ (elements of array manifold vectors will be indexed using superscripts). Let N be the number of microphones in the array. Note that $\mathrm{vec}\left\{\mathbf{v}_{\mathbf{u}_m}\mathbf{v}_{\mathbf{u}_m}^H\right\} = \mathbf{v}_{\mathbf{u}_m}^* \otimes \mathbf{v}_{\mathbf{u}_m}$, where $\otimes$ is the Kronecker product. Therefore, $\mathbf{A} \in \mathbb{C}^{N^2 \times M}$ and

$$\begin{aligned} \mathbf{s} &= \mathbf{A}\mathbf{y} && (7)\\ &= \left[\, \mathbf{v}_{\mathbf{u}_0}^* \otimes \mathbf{v}_{\mathbf{u}_0} \ \ \mathbf{v}_{\mathbf{u}_1}^* \otimes \mathbf{v}_{\mathbf{u}_1} \ \cdots \ \mathbf{v}_{\mathbf{u}_{M-1}}^* \otimes \mathbf{v}_{\mathbf{u}_{M-1}} \,\right]\mathbf{y}. && (8) \end{aligned}$$

Given a two-dimensional array, its array manifold vector $\mathbf{v}(\mathbf{u}) = \mathbf{v}(u_x, u_y)$ is said to be separable if there exist $\mathbf{a}(u_x)$ and $\mathbf{b}(u_y)$ such 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.

Let us specify the enumeration of look directions we will use. Suppose that $\mathbf{Y}$ is an $M_x \times M_y$ acoustic image (note that $\mathbf{Y}$ has M_x columns and M_y rows). The rows of $\mathbf{Y}$ correspond to horizontal scan lines of sampled pixels, and the columns of $\mathbf{Y}$ correspond to vertical scan lines of sampled pixels. Let $\{u_{x_m}\}_{m=0}^{M_x-1}$ and $\{u_{y_n}\}_{n=0}^{M_y-1}$ be points which sample U-space along the x and y axes, ordered from left to right and from top to bottom. We define $\mathbf{u}_0, ..., \mathbf{u}_{M-1}$ such that $\mathbf{y} = \mathrm{vec}\{\mathbf{Y}\}$ and

$$\mathbf{y} = \left[\, |Y(\mathbf{u}_0)|^2 \ \ |Y(\mathbf{u}_1)|^2 \ \cdots \ |Y(\mathbf{u}_{M-1})|^2 \,\right]^T. \tag{9}$$

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

$$\mathbf{u}_m = \left[\, u_{x_{\lfloor m/M_y \rfloor}} \ \ u_{y_{\mathrm{mod}(m, M_y)}} \,\right]^T. \tag{10}$$

Suppose the array is separable. Let $N = N_xN_y$ be the number of array elements. Thus, we can write $\mathbf{v}(\mathbf{u}) = \mathbf{v}(u_x, u_y) = \mathbf{v}_x(u_x) \otimes \mathbf{v}_y(u_y)$, where $\mathbf{v}_x$ and $\mathbf{v}_y$ are $N_x \times 1$ and $N_y \times 1$ manifold vectors.

To save space, we will use the shorthand notation

$$\begin{aligned} \mathbf{v}_x(u_{x_m}) &= \mathbf{v}_{x_m} = \left[\, \mathrm{v}_{x_m}^0 \ \ \mathrm{v}_{x_m}^1 \ \cdots \ \mathrm{v}_{x_m}^{N_x-1} \,\right]^T \\ \mathbf{v}_y(u_{y_n}) &= \mathbf{v}_{y_n} = \left[\, \mathrm{v}_{y_n}^0 \ \ \mathrm{v}_{y_n}^1 \ \cdots \ \mathrm{v}_{y_n}^{N_y-1} \,\right]^T. \end{aligned} \tag{11}$$

Using the separability of the array in (8), we obtain

$$\begin{aligned} \mathbf{A} = \Big[&(\mathbf{v}_{x_0}^* \otimes \mathbf{v}_{y_0}^*) \otimes (\mathbf{v}_{x_0} \otimes \mathbf{v}_{y_0}) \ \cdots \\ &\cdots \ (\mathbf{v}_{x_{M-1}}^* \otimes \mathbf{v}_{y_{M-1}}^*) \otimes (\mathbf{v}_{x_{M-1}} \otimes \mathbf{v}_{y_{M-1}}) \Big] \end{aligned} \tag{12}$$

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

$$\left[\, \mathrm{v}_{x_0}^{i*}\mathrm{v}_{x_0}^{j} \ \cdots \ \mathrm{v}_{x_{M_x-1}}^{i*}\mathrm{v}_{x_{M_x-1}}^{j} \,\right] \otimes \left[\, \mathrm{v}_{y_0}^{k*}\mathrm{v}_{y_0}^{l} \ \cdots \ \mathrm{v}_{y_{M_y-1}}^{k*}\mathrm{v}_{y_{M_y-1}}^{l} \,\right], \tag{13}$$

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

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

$$c_m(i, j) = \mathrm{v}_{x_m}^{i*}\mathrm{v}_{x_m}^{j} \qquad d_n(k, l) = \mathrm{v}_{y_n}^{k*}\mathrm{v}_{y_n}^{l}. \tag{14}$$

For $0 \le m, n < N_xN_y$, 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

$$\begin{aligned} \mathbf{c}(i, j) &= \left[\, c_0(i, j) \ \cdots \ c_{M_x-1}(i, j) \,\right]^T \\ \mathbf{d}(k, l) &= \left[\, d_0(k, l) \ \cdots \ d_{M_y-1}(k, l) \,\right]^T. \end{aligned}$$

Using (13), we have

$$\begin{aligned} \mathbf{S}_{n,m} &= \left[\mathbf{c}^T(i, j) \otimes \mathbf{d}^T(k, l)\right]\mathrm{vec}\{\mathbf{Y}\} && (15)\\ &= \mathbf{d}^T(k, l)\,\mathbf{Y}\mathbf{c}(i, j), && (16) \end{aligned}$$

where $i = \left\lfloor \frac{m}{N_y} \right\rfloor$, $j = \left\lfloor \frac{n}{N_y} \right\rfloor$, $k = \mathrm{mod}(m, N_y)$, $l = \mathrm{mod}(n, N_y)$. Also, (15) and (16) are equivalent because $\left(\mathbf{A}^T \otimes \mathbf{B}\right)\mathrm{vec}\{\mathbf{C}\} = \mathrm{vec}\{\mathbf{BCA}\}$ whenever $\mathbf{BCA}$ is defined [6].

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

$$(i,j) \diamond (k,l) = \mathbf{d}^T (k,l) \mathbf{Y} \mathbf{c} (i,j) \tag{17}$$

and

$$\mathbf{T}_{j,i} = \begin{bmatrix} (i,j) \diamond (0,0) & \cdots & (i,j) \diamond (N_y - 1, 0) \\ (i,j) \diamond (0,1) & \cdots & (i,j) \diamond (N_y - 1, 1) \\ \vdots & & \vdots \\ (i,j) \diamond (0, N_y - 1) & \cdots & (i,j) \diamond (N_y - 1, N_y - 1) \end{bmatrix} \tag{18}$$

From the results above, it can be shown that

$$\mathbf{S} = \begin{bmatrix} \mathbf{T}_{0,0} & \mathbf{T}_{0,1} & \cdots & \mathbf{T}_{0,N_x-1} \\ \mathbf{T}_{1,0} & \mathbf{T}_{1,1} & \cdots & \mathbf{T}_{1,N_x-1} \\ \vdots & \vdots & & \vdots \\ \mathbf{T}_{N_x-1,0} & \mathbf{T}_{N_x-1,1} & \cdots & \mathbf{T}_{N_x-1,N_x-1} \end{bmatrix}. \tag{19}$$

Let

$$\mathbf{t}_{i,j} = \text{vec}\left\{\mathbf{T}_{i,j}\right\} \tag{20}$$

$$\mathbf{Z} = \left[\, \mathbf{t}_{0,0} \; \mathbf{t}_{1,0} \; \ldots \; \mathbf{t}_{N_x-1,N_x-1} \,\right]. \tag{21}$$

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

$$\mathbf{V_x} = \left[\, \mathbf{c}(0,0) \;\; \mathbf{c}(0,1) \;\; \cdots \;\; \mathbf{c}(N_x - 1, N_x - 1) \,\right]^T \tag{22}$$

$$\mathbf{V_y} = \left[\, \mathbf{d}(0,0) \;\; \mathbf{d}(0,1) \;\; \cdots \;\; \mathbf{d}(N_y - 1, N_y - 1) \,\right]^T \tag{23}$$

By comparison with (17), one can verify that

$$\mathbf{Z} = \mathbf{V_y} \mathbf{Y} \mathbf{V_x}^T, \tag{24}$$

which is the fast implementation of $\mathbf{A}$.

We now write $\mathbf{A}$ in terms of the above, which will show why (24) is a fast transform, and allow us to immediately obtain a fast version of $\mathbf{A}^H$. It follows from (24) that $\text{vec}\left\{\mathbf{Z}\right\} = \left(\mathbf{V_x} \otimes \mathbf{V_y}\right) \text{vec}\left\{\mathbf{Y}\right\}$. Define $\boldsymbol{\Xi}$ such that $\text{vec}\left\{\mathbf{S}\right\} = \boldsymbol{\Xi}\text{vec}\left\{\mathbf{Z}\right\}$ (note that $\boldsymbol{\Xi}$ is a permutation). Thus, $\text{vec}\left\{\mathbf{S}\right\} = \boldsymbol{\Xi}\left(\mathbf{V_x} \otimes \mathbf{V_y}\right) \text{vec}\left\{\mathbf{Y}\right\}$ and

$$\mathbf{A} = \boldsymbol{\Xi}\left(\mathbf{V_x} \otimes \mathbf{V_y}\right). \tag{25}$$

Under the simplifying assumption that $N_x = N_y$ and $M_x = M_y$, one can show that multiplication by (25) has cost $O\left(MN^2\right)$, while (24) has cost $O\left(NM + N^2 M^{1/2}\right)$. Since N and $M^{1/2}$ are large, these savings correspond to many orders of magnitude. Also, (24) is easily parallelizable. Finally, in many applications $\boldsymbol{\Xi}$ (which has negigible cost) does not even have to be implemented as part of the transform, as long as one preprocesses the measured $\mathbf{S}$ by $\boldsymbol{\Xi}^{-1}$.

Recall that we introduced $\mathbf{Y}$ as having scan lines which realize an arbitrary separable sampling of U-space. If $\left\{u_{x_m}\right\}$ and $\left\{u_{y_i}\right\}$ uniformly sample U-space, then $\mathbf{V_x}$ and $\mathbf{V_y}$ can be interpreted as DFT matrices for non-uniform 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, an additional optimization for (24) consists of replacing the products by $\mathbf{V_x}$ and $\mathbf{V_y}$ with fast non-equispaced Fourier transforms (NFFTs) [7]. 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 $M_y > 2^8$.

3.2. Fast adjoint transform

Regularized least-squares methods typically require implementations of $\mathbf{A}$ and $\mathbf{A}^H$. A computationally efficient reconstruction algorithm should use fast implementations of both, otherwise the slow transform becomes a bottleneck for the solver. We show below how to obtain a fast version of $\mathbf{A}^H$.

Let $\bar{\mathbf{Y}} \in \mathbb{R}^{M_y \times M_x}$ such that $\text{vec}\left\{\bar{\mathbf{Y}}\right\} = \mathbf{A}^H \text{vec}\left\{\mathbf{S}\right\}$. Since $\boldsymbol{\Xi}$ is a permutation, $\boldsymbol{\Xi}^{-1} = \boldsymbol{\Xi}^T$ and $\text{vec}\left\{\mathbf{Z}\right\} = \boldsymbol{\Xi}^T \text{vec}\left\{\mathbf{S}\right\}$. Using (25), $\text{vec}\left\{\bar{\mathbf{Y}}\right\} = \left(\mathbf{V_x}^H \otimes \mathbf{V_y}^H\right) \text{vec}\left\{\mathbf{Z}\right\}$ and

$$\bar{\mathbf{Y}} = \mathbf{V_y}^H \mathbf{Z} \mathbf{V_x}^*, \tag{26}$$

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

For separable arrays which are uniformly sampled in U-space, multiplication by $\mathbf{V_x}$ and $\mathbf{V_y}$ can again be replaced by NFFTs, under the same considerations presented for the direct transform.

3.3. 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 is of practical interest, as we will use it in Section 4.2 to accelerate the state-of-the-art deconvolution algorithm DAMAS2 [4] by an order of magnitude.

One obtains a fast version of $\text{vec}\left\{\bar{\mathbf{Y}}\right\} = \mathbf{A}^H\mathbf{A}\text{vec}\left\{\mathbf{Y}\right\}$ by evaluating (24) followed by (26). These can be combined as

$$\bar{\mathbf{Y}} = \mathbf{V_y}^H \mathbf{V_y} \mathbf{Y} \mathbf{V_x}^T \mathbf{V_x}^*. \tag{27}$$

When N_x and N_y are sufficiently large with respect to M_x and M_y, the fastest approach is to precompute $\mathbf{V_y}^H\mathbf{V_y}$ and $\mathbf{V_x}^T\mathbf{V_x}^*$ (which are real-valued), saving the cost of computing a relatively large $\mathbf{Z}$ as an intermediate result.

4. APPLICATIONS

Due to limited space, we only present two applications. This transform can also be used to enable fast imaging with state-of-the-art regularized least-squares solvers [8, 9], producing aeroacoustic images with unprecedented accuracy. For more details, we refer the reader to [10].

The following applications are strongly bottlenecked by the transform. Thus, accelerating the transform by a given factor accelerates image reconstruction by nearly the same factor. The relative runtimes for different implementations of $\mathbf{A}$ and $\mathbf{A}^H\mathbf{A}$ are shown in Fig. 1. We also compared this fast transform to our more general (but slower) fast transforms based on the NFFT (fast non-equispaced Fourier transform) and NNFFT (fast non-equispaced in time and frequency Fourier transform), presented in [10]. These allow arbitrary planar array geometries and (for the NNFFT-based transform) arbitrary U-space samplings.

Runtimes were obtained on an Intel Core 2 Duo T9400 processor in 64-bit mode, using only one core. All transforms were implemented in MATLAB R2008b, and the NFFT library (written in C and based on FFTW) was compiled with default optimizations.

4.1. Delay-and-sum imaging

Note that imaging with delay-and-sum (DS) beamforming uses the following (somewhat crude) approximation:

$$\begin{aligned} \left|Y\left(u_{x_m}, u_{y_n}\right)\right|^2 &\approx \mathbf{v}^H\left(u_{x_m}, u_{y_n}\right) \mathbf{S} \mathbf{v}\left(u_{x_m}, u_{y_n}\right) \\ &= (\mathbf{v}_{x_m}^H \otimes \mathbf{v}_{y_n}^H) \mathbf{S} (\mathbf{v}_{x_m} \otimes \mathbf{v}_{y_n}) \end{aligned} \tag{28}$$

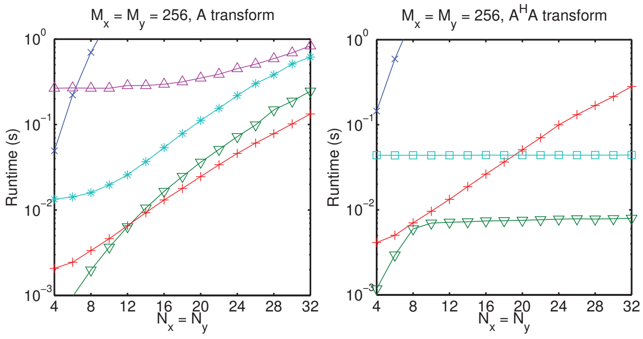

Fig. 1. Runtimes for $\mathbf{A}$ and $\mathbf{A}^H\mathbf{A}$. ×: explicit matrix representation given by (25), ▽: proposed transform, implemented with (24) using matrix products, +: proposed transform, implemented with (24) replacing matrix products with NFFTs, ∗: fast NFFT-based transform for arbitrary geometries [10], △: fast NNFFT-based transform for arbitrary geometries and U-space samplings [10], □: 2D FFT-accelerated convolution. Due to limited space, we only present results for $M_x = M_y = 256$, but note that the proposed transform scales well for different (and larger) problem sizes.

$$= \left[\left(\mathbf{v}_{xm} \otimes \mathbf{v}_{yn}\right) \otimes \left(\mathbf{v}_{xm}^{*} \otimes \mathbf{v}_{yn}^{*}\right)\right] \operatorname{vec}\{\mathbf{S}\}. \tag{29}$$

By definition $\left|Y\left(u_{xm}, u_{yn}\right)\right|^2 = \mathbf{Y}_{n,m}$, and comparing (29) with (12) it follows that

$$\operatorname{vec}\{\mathbf{Y}\} \approx \mathbf{A}^H \operatorname{vec}\{\mathbf{S}\}. \tag{30}$$

Thus, DS imaging can be implemented with $\mathbf{A}^H$.

4.2. DAMAS2

Let $\check{\mathbf{Y}}$ be the image obtained with DS beamforming, $\mathbf{P}$ the array PSF for DS imaging, $\mathbf{Y}$ the clean image and $\hat{\mathbf{Y}}^{(k)}$ the reconstructed image at iteration k. By definition, $\check{\mathbf{Y}} = \mathbf{P} * \mathbf{Y}$, where $*$ represents 2D convolution.

DAMAS2 [4] solves for $\mathbf{Y}$ by iterating

$$\hat{\mathbf{Y}}^{(k+1)} = \max\left\{\hat{\mathbf{Y}}^{(k)} + \frac{1}{a}\left[\check{\mathbf{Y}} - \left(\mathbf{P} * \hat{\mathbf{Y}}^{(k)}\right)\right], \mathbf{0}\right\}, \tag{31}$$

where $\max\{\cdot,\cdot\}$ returns the pointwise maximum, $a = \sum_{i,j}|\mathbf{P}_{i,j}|$, $\hat{\mathbf{Y}}^{(0)} = \mathbf{0}$ and convolution is implemented with a 2D FFT.

It follows from Section 4.1 that $\mathbf{A}^H\mathbf{A}$ obtains the DS image from a clean image. Since the DS image is the clean image convolved with the beamformer PSF, $\mathbf{A}^H\mathbf{A}$ can also be implemented with an FFT-accelerated convolution. As shown in Fig. 1, (27) can always be used to implement $\mathbf{A}^H\mathbf{A}$ more efficiently than an equivalent FFT-accelerated convolution. Implementing $\mathbf{A}^H\mathbf{A}$ with the proposed transform has the added benefit of only requiring a separable (not necessarily uniform) U-space sampling (as opposed to implementing $\mathbf{A}^H\mathbf{A}$ with the 2D FFT, which requires rectangular uniform U-space sampling).

Thus, by using the fast transform it is possible to obtain a faster version of the FFT-accelerated DAMAS2 with

$$\hat{\mathbf{y}}^{(k+1)} = \max\left\{\hat{\mathbf{y}}^{(k)} + \frac{1}{a}\left[\check{\mathbf{y}} - \mathbf{A}^H\mathbf{A}\hat{\mathbf{y}}^{(k)}\right], \mathbf{0}\right\}. \tag{32}$$

Since convolutions are the bottleneck of DAMAS2, the speedup of (32) with respect to (31) is given by the runtime of $\mathbf{A}^H\mathbf{A}$ when compared to that of an FFT accelerated convolution. By referring to Fig. 1, one can see that our proposal outperforms the FFT convolutions by approximately a factor of 10.

5. CONCLUSION

This work presents a fast transform designed to enable computationally efficient and accurate array imaging. To obtain fast implementations, we assumed a separable array geometry, source parameterization in U-space and far-field sources.

We have shown how to apply it to accelerate the DAMAS2 deconvolution approach by a factor of 10, and beamforming by at least two orders of magnitude. It can also be used to enable array imaging with regularized least-squares methods (which would be intractable with explicit matrix multiplications), producing reconstructions in less time than deconvolution approaches, and with greater accuracy. For more details, we refer the reader to [10].

In contrast with FFT- and NFFT-based transforms, the proposed transform allows arbitrary separable samplings of the source distributions, which let one oversample regions with sources and undersample silent areas without performance degradation. The proposed transform makes no numerical approximations, and can be easily implemented and parallelized, since it only requires relatively small matrix products and simple permutations.

Even though this transform was motivated with the far-field assumption, it does not impose any structure onto the array manifold vector other than its separability. The specific representation used in (22) and (23) (which is exact for the far-field case) was used only for convenience, since one can use any other separable representation which is more suitable for the near-field case. This makes the proposed transform more versatile than FFT-based alternatives, as it can be designed to model the effects of near-field propagation. Efficiently combining this idea with calibration matrices is the subject of an upcoming publication.

6. REFERENCES

[1] W.C. Home, K.D. James, T.K. Arledge, P.T. Sodermant, N. Burnside, and S.M. Jaeger, “Measurements of 26%-scale 777 Airframe Noise in the NASA Ames 40- by 80-Foot Wind Tunnel,” in *Proc. of the 11th AIAA/CEAS Aeroacoustics Conference*, 2005.

[2] A. O’Donovan, R. Duraiswami, and D. Zotkin, “Imaging concert hall acoustics using visual and audio cameras,” in *Proc. of ICASSP*, 2008, pp. 5284–5287.

[3] Y. Wang, J. Li, P. Stoica, M. Sheplak, and T. Nishida, “Wideband RELAX and wideband CLEAN for aeroacoustic imaging,” *The Journal of the Acoustical Society of America*, vol. 115, pp. 757, 2004.

[4] R.P. Dougherty, “Extensions of DAMAS and Benefits and Limitations of Deconvolution in Beamforming,” in *Proc. of the 11th AIAA/CEAS Aeroacoustics Conference*, 2005.

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

[6] R.A. Horn and C.R. Johnson, *Matrix analysis*, Cambridge University Press, 1990.

[7] 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, pp. 19, 2009.

[8] 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.

[9] C. Li, “An efficient algorithm for total variation regularization with applications to the single pixel camera and compressive sensing,” M.S. thesis, Rice University, 2009.

[10] F.P. Ribeiro and V.H. Nascimento, “Computationally efficient regularized acoustic imaging,” in *Proc. of ICASSP*, 2011.