Acoustic Imaging Using the Kronecker Array Transform

Vítor H. Nascimento
University of São Paulo

Bruno S. Masiero
University of Campinas

Flávio P. Ribeiro
Microsoft

May 3, 2015

Abstract

The acoustic imaging problem consists of mapping the directions and intensities of sound sources using microphone arrays. These maps are used to design airplanes, cars and trains that are more efficient aerodynamically and less noisy, and also to analyze structures such as concert halls and turbines. In this chapter we describe ways to accelerate the computation of acoustic images, in particular the Kronecker array transform (KAT). We start by giving a short description of the problem of acoustic imaging, and the main state-of-the-art methods for solving it, from the standard beamforming method, through more accurate solutions such as DAMAS2 and covariance-fitting. We proceed by describing the KAT and how it can be applied to accelerate these methods, or to make possible the application of even more powerful methods, such as sparse regularized estimation techniques, which without the KAT would be too computer-intensive to be used in acoustic imaging.

1 Introduction

An *acoustic image* is generated when acoustic levels are coded into a colormap, generating an image of sound level as a function of direction of arrival. Acoustic images are commonly associated with the problem of detecting and characterizing acoustic sources.

Acoustic images can be superimposed over photographs, for instance, to identify unknown sound sources or to compare the relative sound power emitted by a set of sound sources. They may be used for noise reduction and analysis, typically present in the prototyping stages of machine and vehicle development [14], and in the analysis of wind-tunnel measurements [16], turbine noise [22], and in vortex-borne noise detection [4].

Acoustic levels are associated with a sound field, which is an acoustic wave field resulting from the interaction between an acoustic or a vibration source and an elastic medium, such as air or water. Acoustic levels at a point in the sound field can be either directly measured or, when that is not possible, estimated from measurements elsewhere in the same sound field. Such measurements are usually taken using an acoustic sensor array, such as, for example, a microphone or a hydrophone array.

A straightforward method to detect the presence of a sound source is to scan a measurement grid over a closed surface with an intensity probe [10]. If the total average intensity leaving this surface is greater than zero, then there is at least one sound source present inside this surface [17].

If one wishes, however, to more precisely characterize the sound source or even estimate the source's surface velocity out of acoustic measurements, then near-field acoustic holography (NAH) should be used [38]. By adequately sampling a surface containing all of the sources that generate the desired sound field, NAH allows the extrapolation of the field's behavior in other regions of the source-free space or even allows one to identify, separate and characterize the sources that generated the wave field.

As with any wave field, an acoustic wave field can be decomposed into its *active* and *reactive* components [30] and only the active component can transport the radiated acoustic energy far away from the sound source—the far-field. The reactive component is composed of evanescent waves whose energy strongly decay while still in the vicinity of the sound source—the near-field. That is the reason why if complete reconstruction of the sound field near the sound source is desired, near-field measurements have to be conducted. A recently proposed method, however, can image the sound pressure field using laser tomography measurements conducted in the source's far-field [24].

Acoustic imaging, on the other hand, focus on a set of methods that can estimate the sound levels arriving at a point in space from different directions, that is, no attempt is made to estimate the whole sound field. Using these methods, one can verify the presence of sound sources and their directions in relation to a microphone array. The sound level arriving from each direction can be estimated through the use of spatial filters matched to both the array geometry and the source directions, or through the solution of a global optimization problem. Although these methods can be designed to suit several design criteria, they are all based on specific models for the signals received at the microphones, and their performance will depend on how well the models used correspond to reality. The plane wave model is most commonly used and is often adequate for sources in the far field. Under this assumption it is possible to estimate the sound intensity arriving at the microphone array from each source in the sound field.

The geometry of the microphone array will directly influence the quality of the acoustic image obtained. A measurement taken with a microphone array can be understood as a spatial sampling of the sound field. Traditional imaging techniques will pass these sampled signals through a spatial filter that acts as a window function convolved with the impinging sound field [18]. Microphone arrays usually have a reduced number of sensors, which results in window functions with a wide beamwidth and, consequently, in a smeared acoustic image.

Several methods have been proposed to increase image resolution without increasing the number of sensors in the array, either by changing the geometry of the array to reduce sidelobes [13, 40]; applying deconvolution techniques to eliminate the effect of the convolution with the response function [8, 9, 36]; or using regularized optimization [41]. These methods improve results, but have the drawback of increased computational costs as they involve the iterative solution of an optimization problem containing products of vectors with rather large matrices.

Furthermore, any imperfections in microphone positioning and gain will result in a different response function, mismatched to the designed spatial filters, and consequently resulting in errors in the estimated acoustic image. The influence of these imperfections can be countered by the calibration of the array, which will be briefly discussed at the end of the chapter.

This chapter describes three methods for accelerating the calculation of vector-matrix products required for all the above mentioned methods: the *non-equispaced in time and frequency fast Fourier transform* (NNFFT), the *non-equispaced fast Fourier transform* (NFFT), and the new *Kronecker array transform* (KAT). The NNFFT is the most general, but slowest, acceleration method. It can be employed for any array geometry or space parametrization (that is, any choice of directions toward which the array will "look"). The NFFT is faster, but is restricted to a uniformly-sampled choice of look directions. Finally, the KAT is the fastest method, but both the array and the look directions must be organized in a separable geometry (i.e., a possibly non-uniform rectangular grid). The KAT however, unlike the other methods, can be extended to the calculation of acoustic images with sources closer to the microphone array, when some of the far-field approximations are no longer valid [26, 27].

The most important advantage of the acceleration methods, and of the KAT in particular, is that they allow one to use more advanced reconstruction algorithms, such as sparse or regularized methods, for larger problems (i.e., with more microphones and look directions).

2 Signal model

The acoustic imaging techniques discussed in this chapter are all model-based techniques, i.e., they use different strategies to solve an inverse problem based on a wave propagation model. The signal model that will be used throughout the chapter is based on the following assumptions. First, we assume that all sources lay in the far field and thus each wave front that arrives at the microphone array is a plane wave. Second, we assume that the sound intensity at the array is low enough that superposition applies (this is not a restrictive assumption in general). Finally, we assume that all sources are statistically uncorrelated (this assumption is not true in general, but is necessary to keep the problem computationally feasible.)

2.1 Wave propagation

The linearized acoustic wave equation in Cartesian coordinates is [38]

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} = \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2}, \tag{1}$$

where c is the speed of sound in air. The solution to (1) can be decomposed at a point $\boldsymbol{x} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ sufficiently far from all sources as a superposition of plane waves propagating in different directions [32]. A single plane wave traveling along the direction $\boldsymbol{k}$ assumes the form

$$x(t, \boldsymbol{p}) = f(t - \boldsymbol{k}^T\boldsymbol{p}, \boldsymbol{u}) = f(t + \omega\boldsymbol{u}^T\boldsymbol{p}/c, \boldsymbol{u}),$$

where $\boldsymbol{k} = \begin{bmatrix} k_x & k_y & k_z \end{bmatrix}^T$ is the *wavenumber* vector, a vector that points in the direction of propagation of the wave, with magnitude $\|\boldsymbol{k}\|_2 = \omega/c$ ($\|\cdot\|_2$ is the Euclidean norm), and $\boldsymbol{u} = -c\boldsymbol{k}/\omega$ is a unit-length vector that points towards the direction *from which* the wave is arriving at the array (see Figure 1). If the waveform $f(t, \boldsymbol{u})$ has a single frequency, $f(t, \boldsymbol{u}) = F(\omega, \boldsymbol{u})e^{j\omega t}$, then the signal at a point $\boldsymbol{p}$ in space has the form $x(t, \boldsymbol{p}) = X(\omega, \boldsymbol{p})e^{j\omega t}$, with[1]

$$X(\omega, \boldsymbol{p}) = F(\omega, \boldsymbol{u})e^{j\frac{\omega}{c}\boldsymbol{u}^T\boldsymbol{p}} = F(\omega, \boldsymbol{u})e^{-j\boldsymbol{k}^T\boldsymbol{p}}. \tag{2}$$

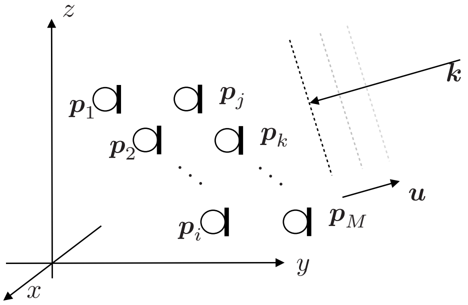

Figure 1: Plane wave and M-element microphone array.

We now use an array with M microphones to sample this sound field. The position of each microphone is given by $\boldsymbol{p}_m = \begin{bmatrix} x_m & y_m & z_m \end{bmatrix}^T$. We write the signal sensed by each microphone when the plane wave arrives at the array from

[1] We remind the reader that model (2) in general is an approximation, valid for a single plane wave in the vicinity of a point $\boldsymbol{p}$. Since it does not consider attenuation or other distortions suffered by the signal, $f(t)$ is *not* the signal actually emitted by the source, but rather the signal *arriving* at point $\boldsymbol{p}$ from a certain direction $\boldsymbol{u}$, using the phase at the origin $\boldsymbol{p} = \boldsymbol{0}$ as a time reference. If the source is sufficiently far, the waveform $f(\cdot)$ will not change on a neighborhood of $\boldsymbol{p}$, only the phase at each point.

direction $\boldsymbol{u}$ (as shown in Figure 1) as

$$\boldsymbol{X}(\omega) = \begin{bmatrix} X(\omega, \boldsymbol{p}_1) \\ X(\omega, \boldsymbol{p}_2) \\ \vdots \\ X(\omega, \boldsymbol{p}_M) \end{bmatrix} = \begin{bmatrix} A_1(\omega, \boldsymbol{u}) e^{j\omega \boldsymbol{u}^T \boldsymbol{p}_1 / c} \\ A_2(\omega, \boldsymbol{u}) e^{j\omega \boldsymbol{u}^T \boldsymbol{p}_2 / c} \\ \vdots \\ A_N(\omega, \boldsymbol{u}) e^{j\omega \boldsymbol{u}^T \boldsymbol{p}_M / c} \end{bmatrix}. \tag{3}$$

$A_m(\omega, \boldsymbol{u}) = G_m(\omega, \boldsymbol{u}) \cdot F(\omega, \boldsymbol{u})$ has a component $G_m(\omega, \boldsymbol{u})$ proportional to the microphone's sensitivity and directivity in addition to the component $F(\omega, \boldsymbol{u})$ relative to the arriving signal. For the time being we assume that all microphones are omnidirectional and that they are all adequately calibrated, so that $G_m(\omega, \boldsymbol{u}) = 1$, and (3) simplifies to

$$\boldsymbol{X}(\omega) = F(\omega, \boldsymbol{u}) \begin{bmatrix} e^{j\omega \boldsymbol{u}^T \boldsymbol{p}_1 / c} \\ e^{j\omega \boldsymbol{u}^T \boldsymbol{p}_2 / c} \\ \vdots \\ e^{j\omega \boldsymbol{u}^T \boldsymbol{p}_M / c} \end{bmatrix} \triangleq F(\omega, \boldsymbol{u}) \boldsymbol{v}(\boldsymbol{u}), \tag{4}$$

where $\boldsymbol{v}(\boldsymbol{u})$ is known as the *array manifold vector.* It contains the relative delays with which the plane wave propagating from direction $-\boldsymbol{u}$ reaches each of the array's sensors. A microphone array will be able to differentiate between plane waves arriving from different directions by the relative delays between the signals at each microphone. The array manifold vector has a fundamental role in acoustic imaging because it is a concise and convenient way of representing these delays, as we show next. Consider the microphone at position $\boldsymbol{p}_1$. Then

$$\boldsymbol{k}^T \boldsymbol{p}_1 = -\frac{\omega}{c} \boldsymbol{u}^T \boldsymbol{p}_1 = -\frac{\omega}{c} \|\boldsymbol{p}_1\|_2 \cos\theta \triangleq \omega \tau_1,$$

where θ is the angle between $\boldsymbol{u}$ and $\boldsymbol{p}_1$, as shown in Figure 2, and τ_1 is the delay at $\boldsymbol{p}_1$, using the phase at the origin as reference, for a wave arriving from direction $-\boldsymbol{u}$. Note that the product of $F(\omega, \boldsymbol{u})$ and each entry of $\boldsymbol{v}(\boldsymbol{u})$ is of the form $e^{-j\omega\tau_i} F(\omega, \boldsymbol{u})$, and thus corresponds to the application of a delay to $f(t)$.

2.2 Superposition of sound sources

If the sound field next to the array is linear and all sources are far enough according to the maximum wavelength and maximum array dimension, the signal received at the microphones will be the superposition of an infinite number of plane waves (for a detailed discussion, see [32]). Of course, in general, a discrete approximation is computed, which corresponds to approximating the signals received at the array as a superposition of a finite number of plane waves coming from certain *previously chosen* directions, as shown in Figure 3. This sampling in $\boldsymbol{u}$-space corresponds to choosing a finite number of directions $\boldsymbol{u}_1 \ldots \boldsymbol{u}_N$

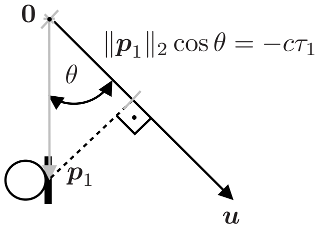

Figure 2: Relative delay for a plane wave arriving from direction $\boldsymbol{k} = -\boldsymbol{u}$. Note that in this example, since the wavefront arrives at $\boldsymbol{p}_1$ before it reaches the origin $\mathbf{0}$, the delay τ_1 is negative.

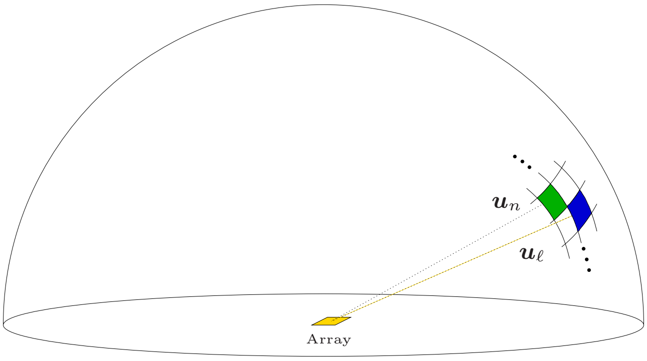

Figure 3: Sampling in $\boldsymbol{u}$-space.

towards which the array will “look”, resulting in a model

$$\boldsymbol{X}(\omega) = \sum_{n=1}^{N} F(\omega, \boldsymbol{u}_n) \cdot \boldsymbol{v}(\boldsymbol{u}_n) = \begin{bmatrix} \boldsymbol{v}(\boldsymbol{u}_1) & \dots & \boldsymbol{v}(\boldsymbol{u}_N) \end{bmatrix} \begin{bmatrix} F(\omega, \boldsymbol{u}_1) \\ \vdots \\ F(\omega, \boldsymbol{u}_N) \end{bmatrix}. \tag{5}$$

We are assuming for the time being that the signals coming from all directions are deterministic. In this case, the (discretized) acoustic image we want to estimate is defined to be the square power for each frequency and direction of the incoming signal, i.e., $Y(\omega, \boldsymbol{u}_n) \triangleq |F(\omega, \boldsymbol{u}_n)|^2$.

We now need to expand our model to include more general kinds of signals, letting the signals $f(t)$ arriving from each direction be stationary random processes. In this case, a direct model like (5) would not be available (since stationary processes do not have finite energy, only the power spectrum is defined). A detailed and precise explanation would take too long, so we refer the reader to [32]. A way around the technical difficulties is to use the discrete Fourier transform (DFT). Define the DFTs of the microphone and source signals over a window of length K, using a proper sampling rate Δt as

$$\hat{F}(\omega_k, \boldsymbol{u}) = \sum_{p=0}^{K-1} f(p\Delta t, \boldsymbol{u}) e^{-j2\pi kp/K}, \tag{6}$$

$$\hat{\boldsymbol{X}}(\omega_k) = \sum_{p=0}^{K-1} \boldsymbol{x}(p\Delta t) e^{-j2\pi kp/K}, \ 0 \leq k \leq K-1, \tag{7}$$

where $\omega_k \triangleq 2\pi p/K$. With these definitions, (5) still holds approximately, with the approximation becoming better as the window length K grows [32].

Define $\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}(\boldsymbol{u}_1) & \dots & \boldsymbol{v}(\boldsymbol{u}_N) \end{bmatrix}$. In this case, the autocorrelation matrix of $\hat{\boldsymbol{X}}(\omega_k)$ can be written as

$$\boldsymbol{R}_x(\omega_k) = \mathrm{E}\{\hat{\boldsymbol{X}}(\omega_k)\hat{\boldsymbol{X}}^H(\omega_k)\} = \boldsymbol{V}\boldsymbol{R}_F(\omega_k)\boldsymbol{V}^H, \tag{8}$$

where $\mathrm{E}\{\cdot\}$ is the expected value, and the source autocorrelation matrix is

$$\boldsymbol{R}_F(\omega_k) = \mathrm{E}\left\{ \begin{bmatrix} \hat{F}(\omega_k, \boldsymbol{u}_1) \\ \vdots \\ \hat{F}(\omega_k, \boldsymbol{u}_N) \end{bmatrix} \begin{bmatrix} \hat{F}^*(\omega_k, \boldsymbol{u}_1) & \dots & \hat{F}^*(\omega_k, \boldsymbol{u}_N) \end{bmatrix} \right\}. \tag{9}$$

Note that the acoustic image corresponds to the diagonal entries of $\boldsymbol{R}_F(\omega_k)$:

$$Y(\omega_k, \boldsymbol{u}_n) = [\boldsymbol{R}_F(\omega_k)]_{n,n} = \mathrm{E}\{|\hat{F}(\omega_k, \boldsymbol{u}_n)|^2\}. \tag{10}$$

In general, $\boldsymbol{R}_F(\omega_k)$ will be a full matrix (meaning that signals arriving from different directions may be correlated). However, if N is large (as we would like it to be, in order to compute an acoustic image with good resolution), taking account of all the $N(N-1)/2$ different correlations would not be feasible, so it is usual to assume that there is no correlation. This is, of course, an approximation, which may create artifacts in the estimated acoustic image.

From now on we omit the frequency ω_k in order to simplify the notation. Under the assumption of uncorrelated signals, the expression for $\boldsymbol{R}_x$ simplifies to

$$\boldsymbol{R}_x = \mathrm{E}\{\hat{\boldsymbol{X}}\hat{\boldsymbol{X}}^H\} = \sum_{n=1}^{N} Y(\boldsymbol{u}_n) \cdot \boldsymbol{v}(\boldsymbol{u}_n)\boldsymbol{v}^H(\boldsymbol{u}_n), \tag{11}$$

so there is a linear relationship between the autocorrelation matrix of the microphone signals (which can be estimated directly from observations) and the desired acoustic image.

This linear relationship becomes more apparent if we rewrite (11) as follows.

For matrices $\boldsymbol{A} = [a_{ij}]$ and $\boldsymbol{B}$ of any dimensions, define

$$\operatorname{vec}(\boldsymbol{A}) \triangleq \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{12} \\ a_{22} \\ \vdots \end{bmatrix}, \qquad \boldsymbol{A} \otimes \boldsymbol{B} \triangleq \begin{bmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \dots \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}.$$

The vec(·) operator therefore corresponds to stacking the columns of a matrix one on top of the other, and the *Kronecker product* $\boldsymbol{A} \otimes \boldsymbol{B}$ of two matrices corresponds to a block matrix in which each block entry is an element of $\boldsymbol{A}$ multiplied by $\boldsymbol{B}$. A property of Kronecker products is that [15]

$$\operatorname{vec}(\boldsymbol{ACB}) = (\boldsymbol{B}^T \otimes \boldsymbol{A}) \operatorname{vec}(\boldsymbol{C}), \tag{12}$$

for any matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ for which the product $\boldsymbol{ACB}$ is defined.

Applying (12) to each term $\boldsymbol{v}(\boldsymbol{u}_n)Y(\boldsymbol{u}_n)\boldsymbol{v}^H(\boldsymbol{u}_n)$ in (11), we obtain

$$\boldsymbol{r}_x \triangleq \operatorname{vec}(\boldsymbol{R}_x) = \underbrace{\begin{bmatrix} \boldsymbol{v}^*(\boldsymbol{u}_1) \otimes \boldsymbol{v}(\boldsymbol{u}_1) & \dots & \boldsymbol{v}^*(\boldsymbol{u}_N) \otimes \boldsymbol{v}(\boldsymbol{u}_N) \end{bmatrix}}_{\triangleq \boldsymbol{A}} \underbrace{\begin{bmatrix} Y(\boldsymbol{u}_1) \\ \vdots \\ Y(\boldsymbol{u}_N) \end{bmatrix}}_{\triangleq \boldsymbol{y}}, \tag{13}$$

where we defined the vector $\boldsymbol{y}$ whose elements are the acoustic image pixels, the vector $\boldsymbol{r}_x$ representing the microphone correlations, and the matrix $\boldsymbol{A}$ that relates them.

2.3 Additive noise

Real measurements are always corrupted by noise, such as thermal noise at the sensors or quantization noise after analog-to-digital conversion. We model the presence of noise as an additive term, replacing (5) by

$$\hat{\boldsymbol{X}} = \sum_{n=1}^{N} \hat{F}(\boldsymbol{u}_n) \cdot \boldsymbol{v}(\boldsymbol{u}_n) + \hat{\boldsymbol{z}}, \tag{14}$$

where $\hat{\boldsymbol{z}}$ is a vector with the additive noise at each microphone at frequency ω_k. We assume that the noise is uncorrelated with our signals of interest and, consequently, the autocorrelation matrix of $\hat{\boldsymbol{X}}$ is updated to

$$\boldsymbol{R}_x = \sum_{n=1}^{N} \left[Y(\boldsymbol{u}_n) \cdot \boldsymbol{v}(\boldsymbol{u}_n)\boldsymbol{v}^H(\boldsymbol{u}_n) \right] + \boldsymbol{R}_z, \tag{15}$$

where $\boldsymbol{R}_z = \mathrm{E}\{\hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^H\}$ is the autocorrelation matrix of the noise component. If the noise is uncorrelated between the sensors, then $\boldsymbol{R}_z = \sigma_z^2 \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix.

3 Methods for acoustic imaging

There are many methods to estimate an acoustic image, with increasing levels of sophistication. All methods mentioned below estimate the sound levels arriving from different directions, assuming the array to be in the far field of all sources.

Common array applications such as antenna arrays, radar and sonar work with narrowband signals. In this case, a single manifold vector matched to the central frequency of the signal can be used to model the plane wave. Acoustic signals, however, are commonly broadband in nature. All of the methods that will be presented in this section calculate the acoustic image using the manifold vector, which depends on ω. Therefore, the acoustic image must be calculated independently for several narrowband-filtered versions of the signal and, if desired, later added together to form a broadband image.

3.1 Spatial filtering

A spatial filter is implemented as a weighted sum of the signals captured by the sensors, such that

$$Z = \boldsymbol{w}^H \hat{\boldsymbol{X}}, \tag{16}$$

where $\boldsymbol{w} = \begin{bmatrix} w_1 & w_2 & \cdots & w_M \end{bmatrix}^T$ is a complex weight vector.

3.1.1 Deterministic beamformer

There are several ways to calculate $\boldsymbol{w}$, the most straightforward manner being deterministic beamformers. In the *Bartlett* beamformer the spatial filter $\boldsymbol{w}$ is chosen so that the filter output power is maximized when the array is excited by a plane wave arriving from $-\boldsymbol{u}$ [21]. Thus, we aim to solve

$$\arg\max_{\boldsymbol{w}} \mathrm{E}\left\{ |Z(\omega)|^2 \right\}. \tag{17}$$

Substituting (14) and (16) into (17) and assuming that the sound field is composed of a single plane wave propagating in direction $-\boldsymbol{u}$, we obtain the cost function

$$J = \left| \boldsymbol{w}^H \boldsymbol{v}(\boldsymbol{u}) \right|^2 R_S + \|\boldsymbol{w}\|^2 \sigma_n^2. \tag{18}$$

To avoid the trivial solution $\|\boldsymbol{w}\| \to \infty$, the Barlett beamformer adds the restriction $\|\boldsymbol{w}\| = 1$ and uses the Cauchy-Schwarz inequality to maximize J, which results in

$$\boldsymbol{w}_{\mathrm{BF}}(\boldsymbol{u}) = \frac{\boldsymbol{v}(\boldsymbol{u})}{\|\boldsymbol{v}(\boldsymbol{u})\|}. \tag{19}$$

Thus, the Bartlett beamformer acts by applying a delay to the signals captured by each sensor, so that the signals arriving from $-\boldsymbol{u}$ are aligned in time and thus constructively added. Note that the Bartlett beamformer is a deterministic method since its weights do not depend on the statistics of the incoming signal, but only on the “listening” direction and the geometry of the array.

Another very common deterministic beamformer is the *delay-and-sum* (DAS) beamformer [32]. Similarly to the Bartlett beamformer, the DAS beamformer seeks to compensate for the relative delay at each sensor and then averages the resulting signals, thus

$$\boldsymbol{w}_{\text{DAS}}(\boldsymbol{u}) = \frac{1}{M}\boldsymbol{v}(\boldsymbol{u}) = \frac{\boldsymbol{v}(\boldsymbol{u})}{\boldsymbol{v}^H(\boldsymbol{u})\boldsymbol{v}(\boldsymbol{u})}. \tag{20}$$

The DAS beamformer is equivalent to the Bartlett beamformer except for a scalar gain.

To obtain the acoustic image, we need to estimate the sound intensity coming from each direction $\boldsymbol{u}$ in a pre-defined grid, to obtain a vector of estimates $\hat{\boldsymbol{y}}$ to $\boldsymbol{y}$. Using a fixed beamformer for each direction in the grid and assuming a perfect estimate of the signal, we have

$$\hat{Y}(\boldsymbol{u}_n) = \mathrm{E}\left\{|\boldsymbol{w}^H(\boldsymbol{u}_n)\hat{\boldsymbol{X}}|^2\right\} = \boldsymbol{w}^H(\boldsymbol{u}_n)\boldsymbol{R}_x\boldsymbol{w}(\boldsymbol{u}_n). \tag{21}$$

The expected value is approximated usually by estimating $\hat{\boldsymbol{X}}$ for a number L of (possibly overlapping) length-K windows, and taking the average of $|\boldsymbol{w}^H\hat{\boldsymbol{X}}|^2$ over the L windows.

Remark that using (12) and (13) with the DAS beamformer gives us,

$$\begin{aligned}\hat{Y}(\boldsymbol{u}_n) &= \frac{1}{M^2}\boldsymbol{v}^H(\boldsymbol{u}_n)\boldsymbol{R}_x\boldsymbol{v}(\boldsymbol{u}_n)\\ &= \frac{1}{M^2}(\boldsymbol{v}^T(\boldsymbol{u}_n)\otimes\boldsymbol{v}^*(\boldsymbol{u}_n))^T\,\mathrm{vec}(\boldsymbol{R}_x) = \frac{1}{M^2}[\boldsymbol{A}]_n^H\boldsymbol{r}_x,\end{aligned} \tag{22}$$

where $[\boldsymbol{A}]_n$ denotes the n-th column of $\boldsymbol{A}$, and thus the DAS beamformer is equivalent to the operation

$$\hat{\boldsymbol{y}} = \frac{1}{M^2}\boldsymbol{A}^H\boldsymbol{r}_x, \tag{23}$$

while the Bartlett beamformer corresponds to

$$\hat{\boldsymbol{y}} = \frac{1}{M}\boldsymbol{A}^H\boldsymbol{r}_x. \tag{24}$$

The artifacts in the image resulting from using these beamformers are better understood thinking in terms of array point-spread functions (PSFs), and their 2-D convolution with the true acoustic image, as we mention further ahead in Section 3.2 [32]. However, it is also useful to compare (23) and (13). We see that $\hat{\boldsymbol{y}}$ may be equal to $\boldsymbol{y}$ only if $\boldsymbol{A}^{-1} = \frac{1}{M^2}\boldsymbol{A}^H$, which would only be true if the columns of $\boldsymbol{A}$ were orthogonal (and $N \leq M^2$ for the inverse to exist). Since in general these conditions are not met, we can expect the image estimated using beamforming to have large artifacts.

3.1.2 Optimal beamformers

The conventional beamformer features a very simple implementation but has the drawback of low image resolution, i.e. if two sound sources are placed too close to each other acoustic images obtained with the conventional beamformer will not be able to resolve both sources. Numerous methods have been proposed as an attempt to improve the image resolution and an important group of such methods are the statistically optimal beamformers. Using the statistics of the sound field, represented by the signals' autocorrelation matrix, these methods can reduce the influence of the neighboring sources resulting in an image with better resolution. As a trade-off these methods are more computationally expensive as they all include the inversion of an autocorrelation matrix as a step to calculate the optimal weights.

In [32, ch. 6] it is demonstrated that for the wide class of optimal beamformers—a class which include the *minimum variance distortionless response* (MVDR) beamformer, the *minimum power distortionless response* (MPDR) beamformer, the *minimum mean square error* (MMSE) beamformer and the *maximum signal to noise* (SNR) beamformer—the resulting spatial filter is an MVDR beamformer followed by a scalar filter dependent on the optimization criterion.

The MVDR beamformer minimizes the variance of the received signal $Z(\omega)$ while keeping a unitary gain at the listening direction, i.e. $\boldsymbol{w}^H\boldsymbol{v}(\boldsymbol{u}) = 1$. This optimization problem can be solve using Lagrange multipliers [32] and results in

$$\boldsymbol{w}^H_{\text{MVDR}}(\boldsymbol{u}) = \frac{\boldsymbol{v}^H(\boldsymbol{u})\boldsymbol{R}_z^{-1}}{\boldsymbol{v}^H(\boldsymbol{u})\boldsymbol{R}_z^{-1}\boldsymbol{v}(\boldsymbol{u})} \tag{25}$$

In practical implementations it may be difficult to estimate $\boldsymbol{R}_z$ if a signal is always present in the direction of interest $\boldsymbol{u}$. In this case, the MPDR beamformer can be used with weights given by

$$\boldsymbol{w}^H_{\text{MPDR}}(\boldsymbol{u}) = \frac{\boldsymbol{v}^H(\boldsymbol{u})\boldsymbol{R}_x^{-1}}{\boldsymbol{v}^H(\boldsymbol{u})\boldsymbol{R}_x^{-1}\boldsymbol{v}(\boldsymbol{u})}. \tag{26}$$

Note that $\boldsymbol{R}_x$ tends to be ill conditioned and Thikonov regularization is usually necessary:

$$\boldsymbol{w}^H_{\text{MPDR}}(\boldsymbol{u}) \approx \frac{\boldsymbol{v}^H(\boldsymbol{u})[\boldsymbol{R}_x + \lambda\boldsymbol{I}]^{-1}}{\boldsymbol{v}^H(\boldsymbol{u})[\boldsymbol{R}_x + \lambda\boldsymbol{I}]^{-1}\boldsymbol{v}(\boldsymbol{u})}, \tag{27}$$

where $\lambda \geq 0$ is the regularization parameter, which must be carefully chosen for good performance.

The acoustic image is generated in the same manner as with the conventional beamforming using (21), that is, $\hat{Y}(\boldsymbol{u}_n) = \text{E}\{|\boldsymbol{w}^H_{\text{MPDR}}(\boldsymbol{u}_n)\boldsymbol{X}|^2\}$. It is not difficult to verify that this is equivalent to

$$\hat{\boldsymbol{y}} = \frac{1}{(\boldsymbol{v}^H(\boldsymbol{u})[\boldsymbol{R}_x + \lambda\boldsymbol{I}]^{-1}\boldsymbol{v}(\boldsymbol{u}))^2}\boldsymbol{A}^H \operatorname{vec}([\boldsymbol{R}_x + \lambda\boldsymbol{I}]^{-1}\boldsymbol{R}_x[\boldsymbol{R}_x + \lambda\boldsymbol{I}]^{-1}).$$

3.2 Deconvolution methods

Consider a single plane wave traveling along direction $-\boldsymbol{u}$. The estimated pixel corresponding to a generic look direction $\boldsymbol{u}_n$ is given by (21). Considering the discrete model (10), we have[2]

$$\hat{Y}(\boldsymbol{u}_n) = Y(\boldsymbol{u}) \cdot \boldsymbol{w}^H(\boldsymbol{u}_n)\boldsymbol{v}(\boldsymbol{u})\boldsymbol{v}^H(\boldsymbol{u})\boldsymbol{w}(\boldsymbol{u}_n) \triangleq Y(\boldsymbol{u}) \cdot P(\boldsymbol{u}_n, \boldsymbol{u}), \tag{28}$$

For a fixed look direction $\boldsymbol{u}_n$, the term $P(\boldsymbol{u}_n, \boldsymbol{u}) = \boldsymbol{w}^H(\boldsymbol{u}_n)\boldsymbol{v}(\boldsymbol{u})\boldsymbol{v}^H(\boldsymbol{u})\boldsymbol{w}(\boldsymbol{u}_n)$, considered as a function of $\boldsymbol{u}$, is the array's *point spread function* (PSF), which describes the gain applied by the array to an input plane wave arriving from direction $-\boldsymbol{u}$ [32]. $P(\boldsymbol{u}_n, \boldsymbol{u})$ is defined over the entire space and can be interpreted as a spatial sampling function that should ideally be maximally sharp, that is (again in our discrete model), equal to

$$P(\boldsymbol{u}_n, \boldsymbol{u}) = \delta_{\boldsymbol{u}_n, \boldsymbol{u}}, \tag{29}$$

where $\delta_{\boldsymbol{u}_n, \boldsymbol{u}} = 1$ if $\boldsymbol{u} = \boldsymbol{u}_n$ and zero otherwise. However, as microphone arrays have a limited number of sensors, their typical PSF will present a larger beamwidth and consequently a smeared acoustic image.

Now, calculating the acoustic image using (21) and considering a superposition of N sources as in (11) results in

$$\hat{Y}(\boldsymbol{u}_n) = \sum_{\ell=1}^{N} Y(\boldsymbol{u}_\ell) \cdot P(\boldsymbol{u}_n, \boldsymbol{u}_\ell). \tag{30}$$

Equation (30) can be interpreted as a spatial convolution [38], i.e. when calculating an acoustic image with conventional or optimal beamformers the result is, in fact, the convolution of the actual acoustic image with the array PSF. This is a second way of explaining the smeared images produced by standard beamformers (compare with (23)).

To reduce the smearing observed in beamforming, several deconvolution techniques have been proposed [8, 9, 29, 36]. They use as inputs the PSF and the image obtained with the DAS beamformer, and generally produce a better approximation of the original source distribution.

3.2.1 DAMAS2

One of the most popular deconvolution methods is the *deconvolution approach for the mapping of acoustic sources* (DAMAS) [3], later improved in [8] and named DAMAS2. Denote by $\boldsymbol{Y}$ the 2-D acoustic image (i.e., $\boldsymbol{y}$ rearranged as a two-dimensional image), and similarly by $\hat{\boldsymbol{Y}}$ the estimated image. DAMAS2 calculates a better approximation $\hat{\boldsymbol{Y}}$ for $\boldsymbol{Y}$ given the DAS estimate (denoted below by $\check{\boldsymbol{Y}}$) by iterating

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

[2] We restrict ourselves again to a discrete spacial distribution of sources, to avoid the long detour necessary to explain adequately the continuous model.

where $*$ denotes 2-D convolution, $\hat{\boldsymbol{Y}}^{(k)}$ is the reconstructed image at iteration k with $\hat{\boldsymbol{Y}}^{(0)} = \boldsymbol{0}$, $\boldsymbol{P}$ is the discretized PSF also arranged as a 2-D array, $a = \sum_{i,j} |\boldsymbol{P}|_{i,j}$, and $\max\{\cdot, \cdot\}$ returns the pointwise maximum. This function is used to guarantee strictly positive power estimates. Convolution can be implemented as a multiplication in the wavelength domain [38] and, therefore, the DAMAS2 algorithm is able to efficiently produce a deconvolved or "clean" acoustic image.

3.3 Covariance fitting

Note that even though DAMAS2 is a state-of-the-art method for computationally efficient acoustic imaging, it does not use any regularization other than forcing pointwise non-negativity, i.e., it does not incorporate a prior model of the source distribution.

Regularized signal reconstruction has been a topic of interest for many decades, and gained significant momentum with the popularity of compressive sensing [5,7]. 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 yet been applied to acoustic imaging.

Considering the presence of noise we rewrite (14) in the matrix form as

$$\boldsymbol{r}_x = \boldsymbol{A}\boldsymbol{y} + \operatorname{vec}\{\boldsymbol{R}_z\} = \boldsymbol{A}\boldsymbol{y} + \sigma^2 \operatorname{vec}\{\boldsymbol{I}\}, \tag{32}$$

assuming spatially uncorrelated noise. Note that the transfer matrix $\boldsymbol{A}$ has usually more columns than rows, so (32) is underdetermined. Prior models of the source distribution can be incorporated as constraints that allow the underdetermined system of equations to be solved.

3.3.1 ℓ_1-regularized least squares

Assume that the acoustic field arriving at the microphone array was generated by only a few compact sources, that is, that the source distribution is *sparse*. In this case we can apply a sparsity constraint to regularize the inversion problem, as suggested in [41], where the following convex optimization problem is proposed

$$\begin{aligned} \underset{\hat{\boldsymbol{y}},\sigma^2}{\text{minimize}} \quad & \left\| \boldsymbol{r}_x - \boldsymbol{A}\hat{\boldsymbol{y}} - \sigma^2 \operatorname{vec}\{\boldsymbol{I}\} \right\|_2^2 \\ \text{subject to} \quad & \hat{\boldsymbol{y}}_{i,j} \geq 0,\ \sigma^2 \geq 0, \text{ and } \|\hat{\boldsymbol{y}}\|_1 \leq \lambda. \end{aligned} \tag{33}$$

The ℓ_1 constraint $\|\hat{\boldsymbol{y}}\|_1 \leq \lambda$ serves to regularize the problem while forcing sparsity. λ is a regularization parameter.

Thanks to the ℓ_1 regularization, the authors of [41] show using numerical examples that by solving (33) one can indeed reconstruct sparse images with very high accuracy. Their proposal outperforms DAMAS2 regarding reconstruction accuracy due to the use of regularization and because no deconvolution was involved.

Another option is to recast (33) as a basis pursuit with denoising problem (BPDN), which has the form

$$\begin{aligned} \underset{\hat{\boldsymbol{y}}}{\text{minimize}} \quad & \|\hat{\boldsymbol{y}}\|_1 \\ \text{subject to} \quad & \|\boldsymbol{r}_x - \boldsymbol{A}\hat{\boldsymbol{y}}\|_2 \leq \sigma. \end{aligned} \tag{34}$$

a kind of optimization problem that has been studied in detail in the compressive sensing literature [35].

3.3.2 Total variation regularized least-squares

To address scenarios where the acoustic images are not sparse in their canonical representations, another possibility is to reconstruct acoustic images with total variation (TV) regularization.

The isotropic total variation norm is defined as

$$\|\boldsymbol{Y}\|_{\text{TV}} = \sum_{i,j} \sqrt{[\nabla_x \boldsymbol{Y}]_{i,j}^2 + [\nabla_y \boldsymbol{Y}]_{i,j}^2} \tag{35}$$

where ∇_x and ∇_y are the first difference operators along the x and y dimensions with periodic boundaries, and i and j are the indices in the x and y dimensions, respectively.

The following optimization problem can then be solved

$$\begin{aligned} \underset{\hat{\boldsymbol{Y}}}{\text{minimize}} \quad & \left\|\hat{\boldsymbol{Y}}\right\|_{TV} + \mu \|\boldsymbol{r}_x - \boldsymbol{A}\hat{\boldsymbol{y}}\|_2^2 \\ \text{subject to} \quad & [\hat{\boldsymbol{Y}}]_{i,j} \geq 0. \end{aligned} \tag{36}$$

The first term measures how much an image oscillates. Therefore, it is smallest for images with plateaus and monotonic transitions, and tends to privilege simple solutions with small amounts of noise. The second term ensures a good fit between the reconstructed image and the measured data. This formulation was first proposed for image denoising [28], and was later generalized and applied successfully to many image reconstruction problems. This method provides accurate and stable image reconstructions with guaranteed convergence.

4 Kronecker array transform

Using covariance-fitting or deconvolution methods, it is possible to obtain acoustic images with good resolution using moderate-sized arrays. These methods are iterative, requiring repeated computation of matrix-vector products of the form $\boldsymbol{A}\hat{\boldsymbol{y}}$ and/or $\boldsymbol{A}^H\hat{\boldsymbol{s}}$. Matrix $\boldsymbol{A}$ is however rather large in practice: for a 64-element array and a 128 × 128-pixel image, matrix $\boldsymbol{A}$ in (13) would be

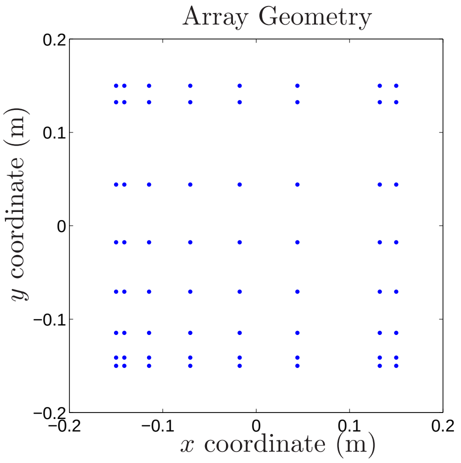

Figure 4: Non-uniform separable geometry. The dots represent positions of microphones.

$64^2 \times 128^2 = 4,096 \times 16,384$, making the more advanced methods very time-consuming. We now show how the structure of $\boldsymbol{A}$ can be used to compute matrix-vector products more efficiently.

There are three strategies for accelerating the computation of acoustic images. When each method can be employed depends on the array geometry (i.e., how the microphones are distributed in space) and on the sampling scheme (i.e., the choice of look directions $\boldsymbol{u}_n$). The NNFFT (non-equispaced in time and frequency fast Fourier transform), applicable to any array geometry or sampling scheme; the NFFT (non-equispaced fast Fourier transform), valid for any *planar* array geometry, and *uniform* sampling of the look directions; and the Kronecker array transform, valid for *planar and separable* array geometries and *separable* sampling of the look directions. By separable we mean that microphones and look directions must be arranged in a rectangular grid, not necessarily uniform, as the example shown in Figure 4.

The KAT provides the largest gain in computational cost, under the constraint of separable geometry and sampling. It can also be combined with the other transforms to further decrease the cost. We describe the three transforms next, additional details can be found in [26].

4.1 NNFFT

Given a sequence of points h_n, the NNFFT is an approximate algorithm for computing expressions of the form [19]

$$\hat{h}_m = \sum_{n=1}^{N} h_n e^{-j2\pi \boldsymbol{b}_n^T \boldsymbol{D} \boldsymbol{c}_m}, \quad 1 \leq m \leq M^2, \tag{37}$$

where $\boldsymbol{D}$ is a diagonal matrix and $\boldsymbol{b}_n$, $1 \leq n \leq N$, $\boldsymbol{c}_m$, $1 \leq m \leq M^2$ are vectors whose entries satisfy

$$-\frac{1}{2} \leq b_{\ell,n} < \frac{1}{2}, \qquad\qquad -C_\ell \leq c_{\ell,m} < C_\ell, \tag{38}$$

for positive constants C_ℓ.

To see that the NNFFT can be used for acoustic imaging, let the entries of $\boldsymbol{u}_n$ be $u_{x,n}$, $u_{y,n}$, and similarly for vector $\boldsymbol{p}_m$. Note that the direction vector is of the form $\boldsymbol{u}_n = \begin{bmatrix} u_{x,n} & u_{y,n} & u_{z,n} \end{bmatrix}^T$, with $u_{x,n}^2 + u_{y,n}^2 + u_{z,n}^2 \leq 1$. Therefore, the entries are in the range $-1 \leq u_{x,n}, u_{y,n}, u_{z,n} \leq 1$.

Take for example the second entry of a product of the form $\hat{\boldsymbol{r}} = \boldsymbol{A}\hat{\boldsymbol{y}}$. The second row of $\boldsymbol{A}$ has the form

$$\begin{aligned} &\begin{bmatrix} v_1^*(\boldsymbol{u}_1)v_2(\boldsymbol{u}_1) & \dots & v_1^*(\boldsymbol{u}_N)v_2(\boldsymbol{u}_N) \end{bmatrix} \\ &= \begin{bmatrix} e^{-j\omega \boldsymbol{u}_1^T(\boldsymbol{p}_1 - \boldsymbol{p}_2)/c} & \dots & e^{-j\omega \boldsymbol{u}_N^T(\boldsymbol{p}_1 - \boldsymbol{p}_2)/c} \end{bmatrix}. \end{aligned}$$

The product of this row by a vector $\hat{\boldsymbol{y}}$ has therefore the form (37), with $\boldsymbol{b}_n = \boldsymbol{u}_n/2$, $\boldsymbol{c}_2 = 2\omega(\boldsymbol{p}_1 - \boldsymbol{p}_2)/(2\pi c)$ and $\boldsymbol{D} = \boldsymbol{I}_3$ (the 3 × 3 identity matrix). The other elements of $\hat{\boldsymbol{r}}$ have similar form, but now with $\boldsymbol{c}_m = \omega(\boldsymbol{p}_i - \boldsymbol{p}_\ell)/(\pi c)$, for a particular pair (i, ℓ) satisfying $1 \leq i, \ell \leq M$. Choosing an ordering of the differences $\boldsymbol{p}_i - \boldsymbol{p}_\ell$, we can use the NNFFT algorithm to compute the product $\hat{\boldsymbol{r}} = \boldsymbol{A}\hat{\boldsymbol{y}}$. The inputs h_n to the NNFFT are the entries of $\hat{\boldsymbol{y}}$ and the outputs $\hat{h}_i$ are the entries of the product $\hat{\boldsymbol{r}}$.

Note that if the array is planar, we can define the coordinate system so that the array lies in the x, y plane, such that $p_{z,m} = 0$ for all microphones. In this case (37) reduces to a two-dimensional transform (similarly, for a linear array only a one-dimensional transform is necessary). Note that in these cases there will be ambiguities between look directions: for example, a planar array cannot distinguish between signals coming from its front or its back.

4.2 NFFT

The NFFT is a faster, but less general algorithm than the NNFFT [19], with a restriction on vector $\boldsymbol{b}_n$ in (37): the entries of $\boldsymbol{b}_n$ must be integers

$$b_{\ell,n} \in \mathbb{Z} : \quad -\frac{N_\ell}{2} \leq b_{\ell,n} < \frac{N_\ell}{2}, \tag{39}$$

for even $N_\ell \in \mathbb{N}$. To satisfy these restrictions, we need to choose adequate look directions $\boldsymbol{u}_n$.

If one is interested in sampling the whole space, one could choose $N = N_x N_y$ with even N_x and N_y, and, in order to obey (39), we would need to choose a uniform sampling:

$$u_{x,n} = \frac{2n_x}{N_x}, \quad -\frac{N_x}{2} \leq n_x < \frac{N_x}{2}, \tag{40a}$$

$$u_{y,n} = \frac{2n_y}{N_y}, \quad -\frac{N_y}{2} \leq n_y < \frac{N_y}{2}. \tag{40b}$$

Defining $b_{x,n} = N_x u_{x,n}/2$, $b_{y,n} = N_y u_{y,n}/2$, we would obey (39) for directions x and y. However, $\boldsymbol{u}_n$ must have unit length, so for direction z we would need $u_{z,n} = \pm\sqrt{1 - u_{x,n}^2 - u_{y,n}^2}$, and the sampling in the z-direction would not be uniform.

A solution is to restrict ourselves to planar arrays, and choose the coordinate system so that the x, y plane corresponds to the array plane. With this choice, the microphone coordinates satisfy $p_{z,m} = 0$ and the $u_{z,n}$ entries vanish in the dot products $\boldsymbol{u}_n^T \boldsymbol{p}_m$. The NFFT algorithm can then be used, with

$$\boldsymbol{b}_n = \begin{bmatrix} \frac{N_x}{2} & 0 & 0 \\ 0 & \frac{N_y}{2} & 0 \end{bmatrix} \boldsymbol{u}_n, \qquad \boldsymbol{c}_m = \frac{\omega}{2\pi c} \begin{bmatrix} \frac{2}{N_x} & 0 & 0 \\ 0 & \frac{2}{N_y} & 0 \end{bmatrix} \boldsymbol{p}_m, \tag{41}$$

The fact that (40) allows $u_{x,n}^2 + u_{y,n}^2 > 1$ means that the transform would compute images for directions that do not in fact exist. This results in a performance loss (since we compute values that we do not need), but for large N_x, N_y there is a net gain. Of course, one could choose $u_{x,n}, u_{y,n}$ restricted to the interval $[-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$, thereby guaranteeing that only values for true look directions are evaluated, but losing information that might come from the edges of the array. See Figure 5.

Acceleration with the FFT

One can verify that, if the microphones are placed in a uniform rectangular grid, the look directions $\boldsymbol{u}_n$ are chosen through uniform sampling of u_x and u_y, and if the frequency of interest ω is such that the distance between consecutive microphones is half the wavelength, then the NFFT just described reduces to the (much faster) FFT. Unfortunately for acoustic imaging this observation is of little use, given that acoustic signals are broadband. In addition, when using covariance fitting methods with regularization, non-uniform microphone arrays lead to better results (see [25] and Section 6).

4.3 Kronecker array transform

Consider a planar array, with microphones placed in a rectangular grid such as that shown in Figure 4. Assume that the coordinate system is chosen so that the array lies in the x, y plane, and that the look directions $\boldsymbol{u}_n$ are chosen such that

$$\boldsymbol{u}_{\ell+(i-1)N_y} = \begin{bmatrix} u_{1,i} \\ u_{2,\ell} \\ \sqrt{1 - u_{1,i}^2 - u_{2,\ell}^2} \end{bmatrix}, \ 1 \le i \le N_x, \ 1 \le \ell \le N_y,$$

where $-1 \le u_{1,1} < u_{1,2} < \cdots < u_{1,N_x} \le 1$, $-1 \le u_{2,1} < u_{2,2} < \cdots < u_{2,N_y} \le 1$. In the notation of the previous sections, we are choosing $u_{x,n} \in \{u_{1,1}, \ldots, u_{1,N_x}\}$,

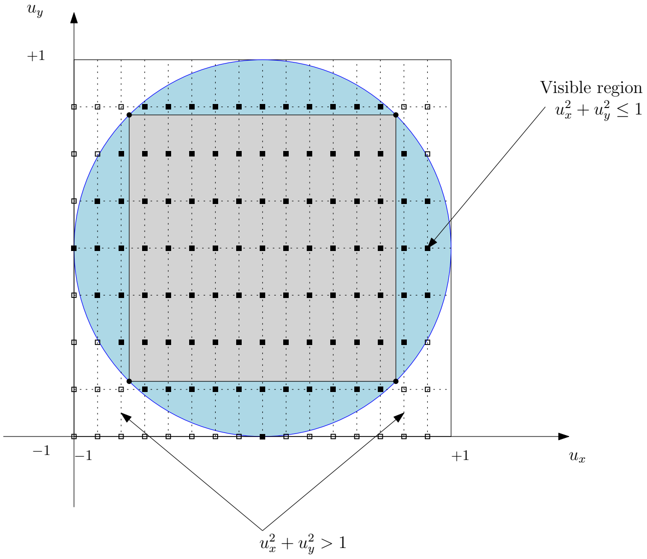

Figure 5: Choice of look directions $\boldsymbol{u}_n$ according to (40), with $N_x = 16$, $N_y = 8$. The blue circle is the "visible" region, for which $u_x^2 + u_y^2 \leq 1$ (black squares). If one wishes to avoid computing unnecessary points with $u_x^2 + u_y^2 > 1$ (white squares), it suffices to choose (u_x, u_y) only inside the gray square, changing the definition of $\boldsymbol{b}_n$ and $\boldsymbol{c}_m$ accordingly. This diagram corresponds to Figure 3 viewed from the top, with the array at the center.

$u_{y,n} \in \{u_{2,1}, \ldots, u_{2,N_y}\}$, so that $u_{x,n} = u_{1,i}$, $u_{y,n} = u_{2,\ell}$ if $n = \ell + (i-1)N_y$. Similarly, the microphone positions are such that

$$\boldsymbol{p}_{s+(r-1)M_y} = \begin{bmatrix} p_{1,r} \\ p_{2,s} \\ 0 \end{bmatrix}, \ 1 \leq r \leq M_x, \ 1 \leq s \leq M_y.$$

In this case, the array manifold vector can be decomposed as follows:

$$\boldsymbol{v}(\boldsymbol{u}_{\ell+(i-1)N_y}) = \boldsymbol{v}_x(u_{1,i}) \otimes \boldsymbol{v}_y(u_{2,\ell}), \tag{42}$$

where

$$\boldsymbol{v}_x(u_{1,i}) \triangleq \begin{bmatrix} e^{j\omega u_{1,i} p_{1,1}/c} \\ e^{j\omega u_{1,i} p_{1,2}/c} \\ \vdots \\ e^{j\omega u_{1,i} p_{1,M_x}/c} \end{bmatrix}, \qquad \boldsymbol{v}_y(u_{2,\ell}) \triangleq \begin{bmatrix} e^{j\omega u_{2,\ell} p_{2,1}/c} \\ e^{j\omega u_{2,\ell} p_{2,2}/c} \\ \vdots \\ e^{j\omega u_{2,\ell} p_{2,M_y}/c} \end{bmatrix}. \tag{43}$$

This can be verified by direct comparison with (4), since

$$e^{j\omega \boldsymbol{u}_n^T \boldsymbol{p}_m/c} = e^{j\omega u_{1,i} p_{1,r}/c} e^{j\omega u_{2,\ell_u} p_{2,s}/c},$$

for $n = \ell + (i-1)N_y$, $m = s + (r-1)M_y$.

Under these conditions, matrix-vector products $\boldsymbol{A}\hat{\boldsymbol{y}}$, $\boldsymbol{A}^H\hat{\boldsymbol{r}}$ and $\boldsymbol{A}^H\boldsymbol{A}\hat{\boldsymbol{y}}$ can be obtained in a more cost-effective way than by direct multiplication. Define the matrices $\boldsymbol{V}_x$ and $\boldsymbol{V}_y$ as follows:

$$\boldsymbol{V}_x \triangleq \begin{bmatrix} v^*_{x,1}(u_{1,1})v_{x,1}(u_{1,1}) & \dots & v^*_{x,1}(u_{1,N_x})v_{x,1}(u_{1,N_x}) \\ v^*_{x,1}(u_{1,1})v_{x,2}(u_{1,1}) & \dots & v^*_{x,1}(u_{1,N_x})v_{x,2}(u_{1,N_x}) \\ \vdots & & \vdots \\ v^*_{x,M_x}(u_{1,1})v_{x,M_x}(u_{1,1}) & \dots & v^*_{x,M_x}(u_{1,N_x})v_{x,M_x}(u_{1,N_x}) \end{bmatrix} \in \mathbb{C}^{M_x^2 \times N_x}, \tag{44}$$

$$\boldsymbol{V}_y \triangleq \begin{bmatrix} v^*_{y,1}(u_{2,1})v_{y,1}(u_{2,1}) & \dots & v^*_{y,1}(u_{2,N_y})v_{y,1}(u_{2,N_y}) \\ v^*_{y,1}(u_{2,1})v_{y,2}(u_{2,1}) & \dots & v^*_{y,1}(u_{2,N_y})v_{y,2}(u_{2,N_y}) \\ \vdots & & \vdots \\ v^*_{y,M_y}(u_{2,1})v_{y,M_y}(u_{2,1}) & \dots & v^*_{y,M_y}(u_{2,N_y})v_{y,M_y}(u_{2,N_y}) \end{bmatrix} \in \mathbb{C}^{M_y^2 \times N_y}. \tag{45}$$

Given a vector $\hat{\boldsymbol{y}}$, define $\hat{\boldsymbol{Y}}$ such that $\hat{\boldsymbol{y}} = \mathrm{vec}(\hat{\boldsymbol{Y}})$, and $\boldsymbol{Z} = \boldsymbol{V}_y\hat{\boldsymbol{Y}}\boldsymbol{V}_x^T$. It can be verified by direct computation that there exists a permutation matrix $\boldsymbol{H}$ such that [26]

$$\boldsymbol{A}\hat{\boldsymbol{y}} = \hat{\boldsymbol{r}} = \boldsymbol{H}\,\mathrm{vec}(\boldsymbol{Z}) = \boldsymbol{H}\,\mathrm{vec}(\boldsymbol{V}_y\hat{\boldsymbol{Y}}\boldsymbol{V}_x^T). \tag{46}$$

This is the Kronecker array transform. Taking advantage of the fact that $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{Y}}$ are real, computing the product $\boldsymbol{A}\hat{\boldsymbol{y}}$ directly requires $0.5M_x^2M_y^2N_xN_y$ complex multiply-and-accumulate (MAC) operations, while using (46) the required number of operations reduces to $0.5M_y^2N_xN_y + M_x^2N_xN_y$ complex MACs if we compute $\boldsymbol{V}_y\hat{\boldsymbol{Y}}$ first, or to $0.5M_x^2N_xN_y + M_x^2M_y^2N_y$ if we compute $\hat{\boldsymbol{Y}}\boldsymbol{V}_x^T$ first.

The products $\hat{\boldsymbol{y}} = \boldsymbol{A}^H\hat{\boldsymbol{r}}$ and $\bar{\boldsymbol{y}} = \boldsymbol{A}^H\boldsymbol{A}\hat{\boldsymbol{y}}$ can be similarly obtained. Define $\bar{\boldsymbol{Z}}$ such that $\mathrm{vec}(\bar{\boldsymbol{Z}}) = \boldsymbol{H}^T\hat{\boldsymbol{r}}$, and $\bar{\boldsymbol{Y}}$ such that $\bar{\boldsymbol{y}} = \mathrm{vec}(\bar{\boldsymbol{Y}})$, then

$$\mathrm{vec}(\boldsymbol{A}^H\hat{\boldsymbol{r}}) = \mathrm{vec}(\boldsymbol{V}_y^H\bar{\boldsymbol{Z}}\boldsymbol{V}_x^*), \qquad \bar{\boldsymbol{Y}} = (\boldsymbol{V}_y^H\boldsymbol{V}_y)\hat{\boldsymbol{Y}}(\boldsymbol{V}_x^T\boldsymbol{V}_x^*). \tag{47}$$

Note that, since $\boldsymbol{V}_y^H\boldsymbol{V}_y$ and $\boldsymbol{V}_x^T\boldsymbol{V}_x^*$ can be pre-calculated, the use of the second form of (47) is more efficient than computing $\boldsymbol{V}_y^H(\boldsymbol{V}_y\hat{\boldsymbol{Y}}\boldsymbol{V}_x^T)\boldsymbol{V}_x^*$.

Simultaneous application of the KAT and NFFT or NNFFT

Since the entries of $\boldsymbol{V}_x$ and $\boldsymbol{V}_y$ are complex numbers with modulus equal to one, the NFFT or the NNFFT can be used to compute the products in (46) and (47), providing further acceleration to the KAT when the number of microphones and of look directions are large enough[3].

[3] Note however that $\boldsymbol{V}_y^H\boldsymbol{V}_y$ and $\boldsymbol{V}_x^T\boldsymbol{V}_x^*$ do not have only entries in the unit circle, so to use the NFFT or the NNFFT to compute $\boldsymbol{A}^H\boldsymbol{A}\hat{\boldsymbol{y}}$, one would need to compute $\boldsymbol{V}_y^H(\boldsymbol{V}_y\hat{\boldsymbol{Y}}\boldsymbol{V}_x^T)\boldsymbol{V}_x^*$.

Transform	Computational cost
KAT with matrix multiplication	$O(MN + M^2N^{1/2})$
KAT with 1-D NFFTs	$O(N \log N + MN^{1/2})$
2-D NFFT	$O(N \log N + M^2)$
2-D NNFFT	$O(N \log N + M^2)$
Matrix multiplication	$O(M^2N)$

Table 1: Asymptotic complexity for different implementations of $\boldsymbol{A}\hat{\boldsymbol{y}}$, assuming $M_x = M_y$ and $N_x = N_y$.

x and y coordinates (m)							
−0.1500	−0.1412	−0.1147	−0.0706	−0.0176	0.0441	0.1324	0.15

Table 2: x and y coordinates of a separable microphone array with 64 microphones (shown in Figure 4). For all microphones, $z = 0$.

5 Computional cost

In Table 1 we list the asymptotic cost of different methods for the case $M_x = M_y$ (i.e., $M = M_x^2$), $N_x = N_y$ ($N = N_x^2$). It is important however to remember that the asymptotic costs in Table 1 do not show the constants multiplying each entry, so from the table one cannot see for example that the NFFT is much faster than the NNFFT.

For a practical application it is also important to consider memory requirements — for example, to directly store $\boldsymbol{A}$, we would need M^2N complex variables; while storing $\boldsymbol{V}_x$ and $\boldsymbol{V}_y$ requires $2MN^{1/2}$ complex variables (again in the case of $M_x = M_y$ and $N_x = N_y$), so using the KAT also reduces memory storage considerably.

The expressions for computational cost give an idea of the advantages of using the KAT, but a full comparison should take into account not only the number of arithmetic operations, but also issues such as memory access and the particular hardware in which the methods are implemented. Figure 6 compares the time required to compute a product $\boldsymbol{A}\hat{\boldsymbol{y}}$, for different dimensions of $\boldsymbol{A}$. The computations were performed on a 64-bit Intel Core 2 Duo T9400 processor using a single core. The permutation $\boldsymbol{H}$ was implemented in ANSI C, the NFFT also used a C implementation with default optimization, as used in [19]. The remaining routines were implemented as m-files in Matlab 2008b.

6 Examples

We present now a few examples of simulated acoustic image reconstructions. In all cases the microphones coordinates $\boldsymbol{p}_m$ are as given in Figure 4. The exact values of the coordinates are given in Table 2.

The examples (following [26]) are simulations of reconstructions of the test images shown in Figure 7. The first test image simulates a sparse source distri-

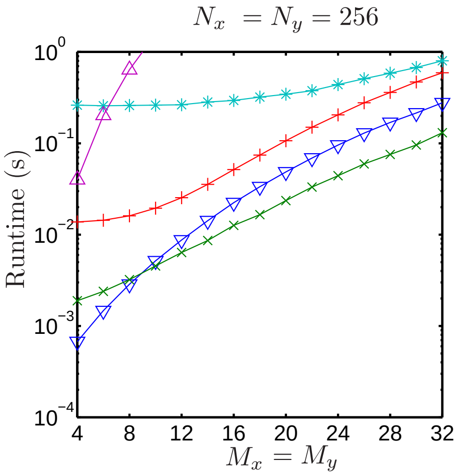

Figure 6: Comparison of runtimes for computation of $\boldsymbol{A}\hat{\boldsymbol{y}}$ in Matlab. ▽: KAT implemented with matrix multiplication; ×: KAT implemented with 1-D NFFTs replacing matrix multiplication; +: direct NFFT implementation ; ∗: direct NNFFT implementation; △: matrix multiplication. From [27].

bution. We compare the results obtained using DAS, DAMAS2, and covariance fitting with ℓ_1 and TV regularization. The regularized optimization problems were solved using package routines SPGL1 [35] for solving (34), using 200 iterations, with $\sigma = 0.01\|\boldsymbol{R}_x\|_F$[4]. For the solution of (36) we used TVAL3 [23] using 100 iterations with $\mu = 10^3$. DAMAS2 used 1,000 iterations. The simulations were performed assuming an 8×8 microphone array, with microphone positions as in Figure 4 and Table 2 ($M_x = M_y = 8$). The look directions were sampled uniformly in u_x, u_y, using the entire range $[-1, 1)$, with 256 points in each direction ($N_x = N_y = 256$). Note that this means that the algorithms compute values also outside the visible region (i.e., for points with $u_x^2 + u_y^2 > 1$). A good test of the quality of reconstruction is to see only small (blue) values outside the visible region. In the next figures, the visible region boundary ($u_x^2 + u_y^2 = 1$) is marked with a black circle. The signals at the microphones were simulated using the ideal image and model (5), assuming a frequency of 6 kHz, and a noise variance set for an SNR of 20 dB.

The results for the sparse pattern can be seen in Figure 8, and for the smoother pattern, in Figure 9. As expected, ℓ_1 regularization has the best performance for the sparse pattern, while TV regularization gives the best results for the smoother pattern.

Finally, we present a comparison of results obtained using separable array geometries with those obtained using a multi-arm logarithmic spiral geometry, specially designed to reduce the sidelobes in the PSF [33], see Figure 10. As can be seen in Figure 11, the result obtained with delay-and-sum algorithm using a

[4] $\|\cdot\|_F$ is the Frobenius norm.

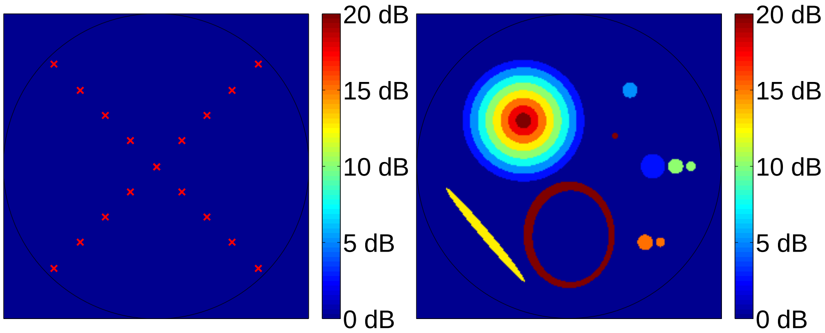

Figure 7: Test images. On the left, a sparse pattern, and on the right, a smoother test image.

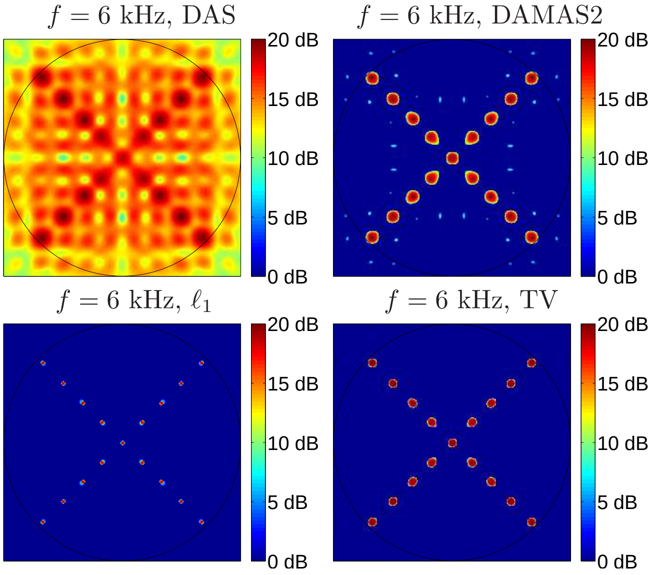

Figure 8: Results for (clockwise from top left) Delay-and-sum, DAMAS2, ℓ_1 regularization and TV regularization, for the sparse pattern. From [26]

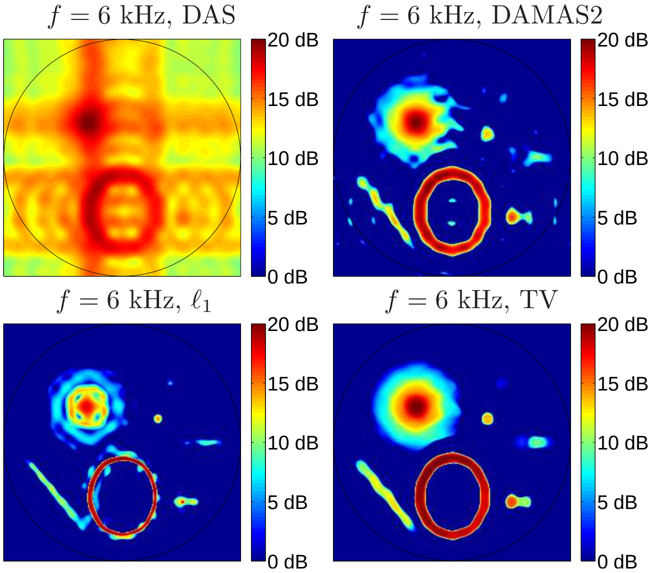

Figure 9: Results for (clockwise from top left) Delay-and-sum, DAMAS2, ℓ_1 regularization and TV regularization, for the smooth pattern. From [26].

50 cm, 63-element logarithmic spiral array is indeed better, compared with the delay-and-sum reconstruction in Figure 9. However, the results obtained with the other methods are comparable.

7 Practical considerations

All methods presented in section 3 assumed perfect knowledge of the relative position of the sensors in the array, and that these sensors present the same sensitivity response in terms of gain and phase. However, depending on how the array is constructed, exact positioning of the microphones cannot be guaranteed. Furthermore, microphones present a variation in their sensitivity response, even when using microphones of the same model.

It is been shown that both variations in gain and phase, as well as microphone positioning errors, result in distortions at the observed source's direction and sound level [20]. Several techniques have been described for calibrating the microphone location. More conventional techniques require knowledge of the exact position of the reference sources used for the calibration [2, 34]. Simultaneous calibration of the position and sensitivity of the microphones has also been analyzed previously [31, 39].

There have been attempts to conduct calibration without prior knowledge of the reference sources [11, 37]. This *blind* calibration algorithms first estimate the position of the reference sources using some "direction-of-arrival" (DOA) algorithm and assuming the nominal array configuration. Next, assuming the

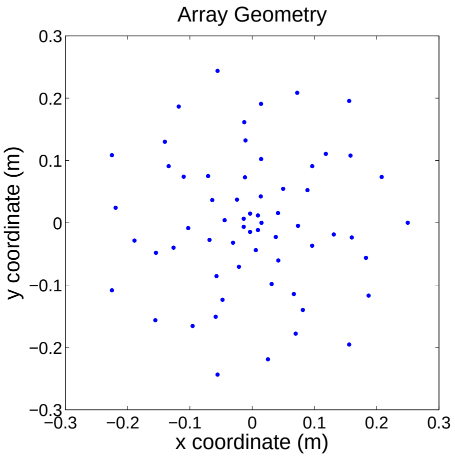

Figure 10: Multi-arm logarithmic spiral array, 50 cm in diameter, with 63 elements.

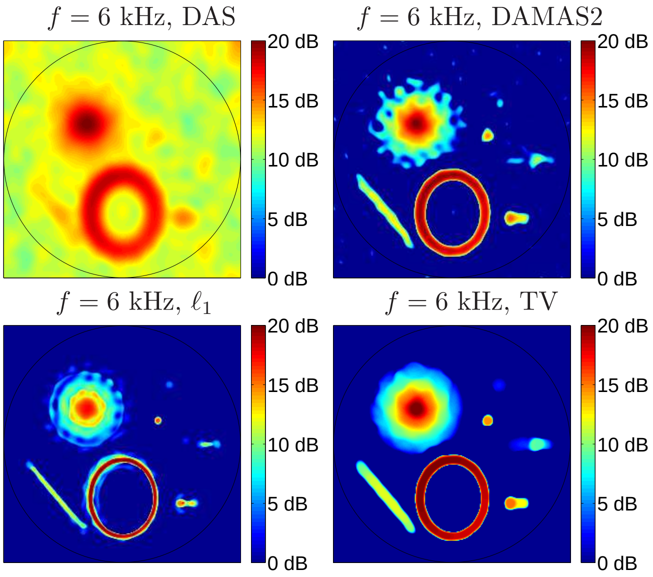

Figure 11: Results for (clockwise from top left) Delay-and-sum, DAMAS2, ℓ_1 regularization and TV regularization, for the smooth pattern using the logarithmic spiral array of Figure 10. One can see, comparing with Figure 9, that the results here are better for DAS, but equivalent for the other methods. From [26].

estimated directions of arrival are correct, the array parameters are estimated. These algorithms repeat these two steps iteratively until the estimate converge. This can be viewed as a joint estimation problem using "group alternating maximization" [32].

Recently, the use of sparsity has been incorporated to the calibration procedure [6]. Simulations demonstrate the effectiveness of the compressive sensing approach to the calibration of even highly uncalibrated measures, when a sufficient number of (unknow, but sparse) signals is provided [1, 12].

8 Conclusion

This chapter provides an introduction to acoustic imaging, describing the main models and assumptions used in the field, and comparing some of the most important algorithms available in the literature. Since the computation of acoustic images is a somewhat computationally demanding task, we emphasized recent methods for speeding up computations, taking advantage of the structure of the array manifold vectors in the far field. The three methods, in order of least to highest acceleration, are the non-equispaced in time and frequency fast Fourier transform (NNFFT), non-equispaced fast Fourier transform (NFFT), and the Kronecker array transform (KAT). Although not described in this chapter, the KAT can also be extended to work when some of the far-field approximations are no longer valid, as described in [27].

References

[1] Bilen, C., Puy, G., Gribonval, R., and Daudet, L. Blind phase calibration in sparse recovery. In *EUSIPCO - 21st European Signal Processing Conference* (Marakech, Morocco, 2013).

[2] Birchfield, S., and Subramanya, A. Microphone Array Position Calibration by Basis-Point Classical Multidimensional Scaling. *IEEE Transactions on Speech and Audio Processing 13* (2005).

[3] Brooks, T. F., and Humphreys, W. M. A Deconvolution Approach for the Mapping of Acoustic Sources (DAMAS) Determined from Phased Microphone Arrays. *Journal of Sound and Vibration 294*, 3 (2004), 856–879.

[4] Brusniak, L., Underbrink, J., and Stoker, R. Acoustic Imaging of Aircraft Noise Sources Using Large Aperture Phased Arrays. *AIAA Journal 2715* (2006), 2006.

[5] Candès, E. J., and Wakin, M. An Introduction To Compressive Sampling. *IEEE Signal Processing Magazine 25*, 2 (2008), 21–30.

[6] Cevher, V., and Baraniuk, R. Compressive sensing for sensor calibration. *2008 5th IEEE Sensor Array and Multichannel Signal Processing Workshop* (2008).

[7] Donoho, D. Compressed sensing. *IEEE Transactions on Information Theory 52*, 4 (Apr. 2006), 1289–1306.

[8] Dougherty, R. Extensions of DAMAS and benefits and limitations of deconvolution in beamforming. *AIAA paper 2961*, 11 (2005).

[9] Ehrenfried, K., and Koop, L. Comparison of iterative deconvolution algorithms for the mapping of acoustic sources. *AIAA Journal 45*, 7 (2007), 1584.

[10] Fahy, F. J. *Sound Intensity.* Elsevier, London, 1987.

[11] Flanagan, B. P., and Bell, K. L. Array self-calibration with large sensor position errors. *Signal Processing 81*, 10 (Oct. 2001), 2201–2214.

[12] Gribonval, R., Chardon, G., and Daudet, L. Blind calibration for compressed sensing by convex optimization. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* (2012), pp. 2713–2716.

[13] Högbom, J. Aperture synthesis with a non-regular distribution of interferometer baselines. *Astronomy and Astrophysics Supplement Series 15* (1974), 417.

[14] Home, W., James, K., Arledge, T., Sodermant, P., Burnside, N., and Jaeger, S. Measurements of 26 percent-scale 777 Airframe Noise in the NASA Ames 40- by 80-Foot Wind Tunnel. In *11 th AIAA/CEAS Aeroacoustics Conference(26 th Aeroacoustics Conference)* (2005), pp. 1–19.

[15] Horn, R. A., and Johnson, C. R. *Matrix Analysis.* Cambridge University Press, 1987.

[16] Humphreys, W., and Brooks, T. Noise spectra and directivity for a scale-model landing gear. *International Journal of Aeroacoustics 8*, 5 (2009), 409–443.

[17] Jacobsen, F. Intensity Probe. In *Handbook of Signal Processing in Acoustics*, D. Havelock, S. Kuwano, and M. Vorländer, Eds. Springer New York, New York, USA, 2008, ch. 58.

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

[19] Keiner, J., Kunis, S., and Potts, D. Using nfft 3—a software library for various nonequispaced fast fourier transforms. *ACM Transactions on Mathematical Software (TOMS) 36*, 4 (2009), 19.

[20] Khong, A., and Brookes, M. The Effect of Calibration Errors on Source Localization with Microphone Arrays. *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07 1* (2007).

[21] Krim, H., and Viberg, M. Two decades of array signal processing research: the parametric approach. 67–94.

[22] Lee, S. Phased-array measurement of modern regional aircraft turbofan engine noise. *AIAA Journal 2653* (2006).

[23] Li, C. An efficient algorithm for total variation regularization with applications to single pixel camera and compressive sensing. Master's thesis, Rice University, 2009.

[24] Oikawa, Y. Spatial Information of Sound Fields. In *Handbook of Signal Processing in Acoustics*, D. Havelock, S. Kuwano, and M. Vorländer, Eds., 2nd ed. Springer New York, New York, USA, 2008, ch. 76, pp. 1403—-1421.

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

[26] Ribeiro, F. P., and Nascimento, V. H. Fast transforms for acoustic imaging— part I: Theory. 2229–2240.

[27] Ribeiro, F. P., and Nascimento, V. H. Fast transforms for acoustic imaging—part II: Applications. 2241–2247.

[28] Rudin, L. I., Osher, S., and Fatemi, E. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena 60* (1992), 259–268.

[29] Sijtsma, P. CLEAN Based on Spatial Source Coherence. In *13th AIAA/CEAS Aeroacoustics Conference (28th AIAA Aeroacoustics Conference)*, Aeroacoustics Conferences. American Institute of Aeronautics and Astronautics, May 2007.

[30] Stanzial, D. Reactive acoustic intensity for general fields and energy polarization. *J. Acoust. Soc. Am. 99*, 4 (1996), 1868.

[31] Tashev, I. Gain self-calibration procedure for microphone arrays. *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763) 2* (2004).

[32] Trees, H. L. V. *Optimum Array Processing*. Wiley, 2002.

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

[34] Valente, S., Tagliasacchi, M., Antonacci, F., Bestagini, P., Sarti, A., and Tubaro, S. Geometric calibration of distributed microphone arrays from acoustic source correspondences. *Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on* (2010).

[35] van den Berg, E., and Friedlander, M. Probing the Pareto frontier for basis pursuit solutins. *SIAM Journal on Scientific Computing 31*, 2 (2008), 890–912.

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

[37] Weiss, A. J., and Friedlander, B. Eigenstructure methods for direction finding with sensor gain and phase uncertainties. *Circuits, Systems, and Signal Processing 9* (1990), 271–300.

[38] Williams, E. G. *Fourier Acoustics: sound radiation and nearfield acoustical holography.* Academic Press, 1999.

[39] Xiao, H., Shao, H.-z., and Peng, Q.-c. A New Calibration Method for Microphone Array with Gain , Phase , and Position Errors. *JOURNAL OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA 5*, 3 (2007), 248–251.

[40] Yan, S., Ma, Y., and Hou, C. Optimal array pattern synthesis for broadband arrays. *The Journal of the Acoustical Society of America 122* (2007), 2686.

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