Computational ultrasound imaging through an aberrating layer using delay-and-sum

De Carlo, Francesca; Brown, Michael; Leus, Geert; Kruizinga, Pieter

DOI
10.23919/EUSIPCO63174.2024.10714937

Publication date
2024

Document Version
Final published version

Published in
2024 32nd European Signal Processing Conference (EUSIPCO)

Computational ultrasound imaging through an aberrating layer using delay-and-sum

Francesca De Carlo*, Michael Brown†‡, Geert Leus§ and Pieter Kruizinga†§

*Fugro, Nootdorp, The Netherlands
†Department of Neuroscience, CUBE, Erasmus MC, Rotterdam, The Netherlands
‡Department of Medical Physics and Biomedical Engineering, University College London, London, United Kingdom
§Department of Microelectronics, Delft University of Technology, Delft, The Netherlands

***Abstract*—Computational ultrasound imaging (cUSi) offers high-resolution 3D imaging with simpler hardware by relying on computational power. Central to cUSi is a large model matrix that stores all pulse-echo signals. For 3D imaging this matrix easily surpasses 1 terabyte, hindering in-memory storage and real-time processing. This paper presents a solution for cUSi through an aberrating layer by introducing a virtual array concept, which uses transfer functions to map data from the real to a virtual array, enabling the use of conventional reconstruction techniques like delay-and-sum (DAS). We demonstrate the mathematical similarity of this approach to using a full model matrix and validate it with promising imaging results.**

***Index Terms*—Computational ultrasound, imaging, virtual array, delay-and-sum**

I. Introduction

Computational ultrasound imaging (cUSi) is a technique of making images from indirect measurements using computationally heavy models that can translate these measurements into images. The advantage of this technique is that it allows for cheaper, simpler, faster and/or problem-specific imaging devices at the expense of significant computing.

The indirect measurements allow, for example, the encoding of spatial information unto a lower dimensional received signal, thereby bypassing the need for Nyquist sampling the spatial domain. Early examples of this approach include the use of reverberant media to reduce the number of sensors required for imaging [1], [2].

In our work we explored the use of a coded aperture mask positioned in front of a receiver to provide the indirect measurements of the spatial information contained in the received echo field. This topology allowed us to perform 3D imaging using a rotating mask in front of a single transceiver [3], high resolution microscopy using a single transceiver and mask with overlapping scan beams [4], and recently 4D imaging of brain hemodynamics in mice using 64 transceivers and a coding aperture mask [5]. We also showed that the design of the coded aperture mask can be optimally tailored to a specific imaging scenario [6].

At the heart of cUSi lays a computational model that describes the pulse-echo measurements of each pixel/voxel in the image. Typically, this is done using a large two-dimensional matrix where each column represents one image pixel and the number of non-zero rows for that column are the signals attributed to that pixel. In typical cUSi systems, these signals are attributed to the complex physical interactions within the 'encoding medium' such as the reverberant cavity or our coding mask. Because of this complexity they cannot be deduced from simple geometrical calculations as is done in classical array processing algorithms such as delay-and-sum (DAS) [7]. Especially for 3D ultrasound imaging when multiple transmissions are needed for one image, this model matrix tends to grow uncomfortably large in terms of size and memory even when a low number (< 100) of transceivers are considered. Practically this means that storing this matrix in memory on, for example, a GPU is not possible and therefore real-time image reconstruction becomes difficult to achieve.

In this paper we explore a solution for this large model matrix problem by mapping the received complex data unto a regular, well-sampled virtual array that is positioned between the homogeneous medium and the 'encoding medium' in front of our transceiver. This 'mapping' process transforms the original data into data as it would be measured with a well-sampled array within the homogeneous medium. This transformation is achieved by using a much smaller matrix containing the transfer functions from every transceiver in the real array to every transceiver in the virtual array. This concept draws inspiration from earlier work by Tanter et al. where an array of 'control points' were considered behind an aberrating skull [8]. After this mapping process the data can be processed using conventional methods such as the DAS algorithm in order to form an image which is almost identical to the one obtained with a full model matrix. Below we introduce this virtual array concept, we show that the method is mathematically similar to when using a full model matrix and provide compelling imaging results to show the validity of the method.

II. Preliminaries

A. Signal model

Consider an array of M elements and an imaging area of N pixels. We define $y_{i,j}(t)$ as the measured signal at sensor j when sensor i transmits, which can be written as

$$y_{i,j}(t) = \sum_{n=1}^{N} [g_{i,n}(t) * g_{j,n}(t)] x_n. \tag{1}$$

Here x_n is the intensity of the scattering at pixel n and $g_{i,n}(t)$ is the Green's function from sensor i to pixel n. Note that in this model we make use of the reciprocity principle of Green's functions and ignore multiple scattering as well as additive noise. We assume we have access to every individual measurement $y_{i,j}(t)$ by activating one element at the time, i.e., a total of M scans is performed where every scan leads to M received signals. This is the so-called synthetic aperture (SA) acquisition scheme. Other schemes can be deduced from this model.

In the frequency domain, (1) becomes

$$Y_{i,j}(\omega) = \sum_{n=1}^{N} G_{i,n}(\omega) G_{j,n}(\omega) x_n. \tag{2}$$

Stacking $Y_{i,j}(\omega)$ over all M elements as $\mathbf{y}(\omega) = [Y_{1,1}(\omega), \ldots, Y_{1,M}(\omega), Y_{2,1}(\omega), \ldots, Y_{M,M}(\omega)]^{\mathrm{T}}$, we obtain

$$\mathbf{y}(\omega) = [\mathbf{G}(\omega) \circ \mathbf{G}(\omega)]\mathbf{x},$$

where $\circ$ is the Khatri-Rao product, $\mathbf{x} = [x_1, \ldots, x_N]^{\mathrm{T}}$ and $[\mathbf{G}(\omega)]_{i,n} = G_{i,n}(\omega)$.

Discretizing the ω domain in L frequencies ω_l, $l = 1, \ldots, L$, and stacking the different $\mathbf{y}(\omega_l)$ vectors into $\mathbf{y} = [\mathbf{y}^{\mathrm{T}}(\omega_1), \ldots, \mathbf{y}^{\mathrm{T}}(\omega_L)]^{\mathrm{T}}$, we finally obtain the full model

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \tag{3}$$

with

$$\mathbf{A} = \begin{bmatrix} \mathbf{G}(\omega_1) \circ \mathbf{G}(\omega_1) \\ \vdots \\ \mathbf{G}(\omega_L) \circ \mathbf{G}(\omega_L) \end{bmatrix}. \tag{4}$$

The measurements can also be arranged in L matrices of dimension $M \times M$. The matrix for frequency ω_l is defined as

$$\mathbf{Y}(\omega_l) = \begin{bmatrix} Y_{1,1}(\omega_l) \ldots Y_{M,1}(\omega_l) \\ \vdots \qquad\qquad \vdots \\ Y_{1,M}(\omega_l) \ldots Y_{M,M}(\omega_l) \end{bmatrix}. \tag{5}$$

Then, the relation between each $\mathbf{Y}(\omega_l)$ and $\mathbf{x}$ is:

$$\mathbf{Y}(\omega_l) = \mathbf{G}(\omega_l)\mathrm{diag}(\mathbf{x})\mathbf{G}^{\mathrm{T}}(\omega_l). \tag{6}$$

In this thesis, we will use the expression in (3) and the one (6) interchangeably.

B. Image formation methods

In this section, we briefly describe the DAS technique and discuss the main idea of model-based imaging.

The most basic beamforming method in ultrasound imaging is DAS. The DAS estimate for pixel n is computed as

$$\hat{x}_n = \sum_{j=1}^{M} \sum_{i=1}^{M} y_{i,j}(t_0 + \tau_{j,n} + \tau_{i,n}) \tag{7}$$

where $\tau_{i,n}$ is equal to the time of flight from sensor i to point n. Since the start of the acquisition may not correspond with the peak of the transmitted pulse, the offset t_0 is introduced to correct the delays [7]. It is clear that DAS, which is based on the geometry of the problem, only works for homogeneous media.

Model-based imaging, on the other hand, can be applied for both homogeneous and inhomogeneous media. In that case, the image is basically generated by solving the system of equations $\mathbf{y} = \mathbf{A}\mathbf{x}$. Focusing for instance on a least squares cost and a two-norm regularizer, this would lead to the estimate

$$\hat{\mathbf{x}} = (\mathbf{A}^{\mathrm{H}}\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{A}^{\mathrm{H}}\mathbf{y}, \tag{8}$$

where λ is the regularizer weight.

To reduce the computational cost, we can also consider the matched filter estimator:

$$\hat{\mathbf{x}} = \mathbf{A}^{\mathrm{H}}\mathbf{y}. \tag{9}$$

For homogeneous media, this matched filter estimator turns out to be closely related to the DAS beamformer. However, it can be used in more general settings compared to DAS.

III. Virtual Array

Starting from the general model, we derive the model that will be used in the rest of the paper. This model is based on the assumption that the inhomogeneities are limited to one layer, situated between the sensor array and the imaging target. In that case, we can assume a virtual array is placed just after the layer, as depicted in Figure 1.

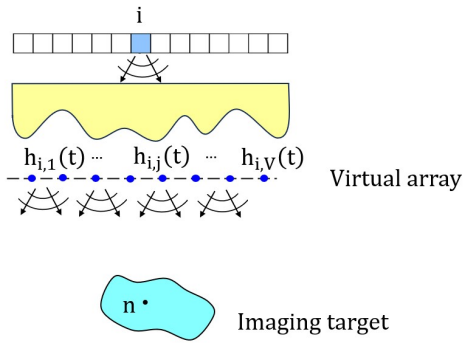

Fig. 1. The virtual array model: The field at a point n in the imaging domain is derived as a function of the pressure on the virtual array, which is measured by the transfer functions $h_{i,j}(t)$.

We define $h_{i,j}(t)$ as the Green's function from element i on the real array to point j on the virtual array. We further define $\tilde{g}_{j,n}(\omega)$ as the Green's function from virtual element j to pixel n. Then, the Green's function from sensor i to pixel n can be written as:

$$g_{i,n}(t) = \sum_{j=1}^{V} h_{i,j}(t) * \tilde{g}_{j,n}(t), \tag{10}$$

where V is the size of the virtual array. Note that this equation holds if the virtual array is Nyquist sampled and the aperture is wide enough to capture all the waves emitted by the real array. The matrix $\mathbf{G}(\omega)$ from Section II can now be written as

$$\mathbf{G}(\omega) = \mathbf{H}(\omega)\tilde{\mathbf{G}}(\omega), \tag{11}$$

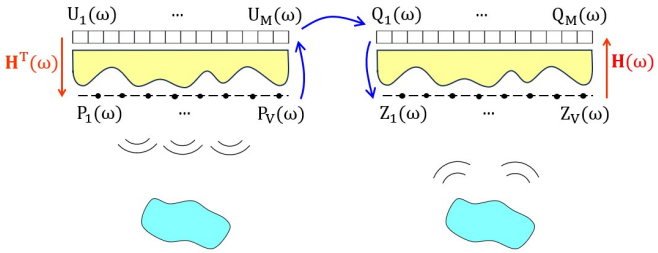

Fig. 2. In the estimation process, we follow the order indicated by the blue arrows. First, we assign values to $\mathbf{p}(\omega)$ and estimate the corresponding input $\mathbf{u}(\omega)$. Next, we derive the real array measurements $\mathbf{q}(\omega)$, from which we estimate the received echoes at the virtual array $\mathbf{z}(\omega)$.

where $[\mathbf{H}(\omega)]_{i,j} = H_{i,j}(\omega)$ and $[\tilde{\mathbf{G}}(\omega)]_{j,n} = \tilde{G}_{j,n}(\omega)$.

Since the region below the layer is homogeneous, $\tilde{\mathbf{G}}(\omega)$ can be derived based on the location of the virtual array and the imaging domain. Thus, $\tilde{\mathbf{G}}(\omega)$ does not depend on the layer and can possibly be inverted by simple geometry-based techniques such as DAS. The presence of the layer only affects $\mathbf{H}(\omega)$.

Based on (11), the model in (6) can be written as:

$$\mathbf{Y}(\omega_l) = \mathbf{H}(\omega_l)\tilde{\mathbf{G}}(\omega_l)\text{diag}(\mathbf{x})\tilde{\mathbf{G}}^{\mathrm{T}}(\omega_l)\mathbf{H}^{\mathrm{T}}(\omega_l). \tag{12}$$

Note that this model does not take into account that the transducers will receive reflections from the layer and hence might be inaccurate.

IV. From the Real to the Virtual Array

We start by defining the notation used in this section (see Figure 2):

- $U_i(\omega)$ is the transmit field at element i of the real array, for $i = 1, .., M$
- $P_j(\omega)$ is the transmit field at element j of the virtual array, for $j = 1, .., V$
- $Z_j(\omega)$ represents the received echoes at element j of the virtual array, for $j = 1, ., V$
- $Q_i(\omega)$ represents the received echoes at element i of the real array, for $i = 1, .., M$

We furthermore stack these variables in the respective vectors $\mathbf{u}(\omega)$, $\mathbf{p}(\omega)$, $\mathbf{z}(\omega)$, and $\mathbf{q}(\omega)$. The relations between these vectors are $\mathbf{p}(\omega) = \mathbf{H}^{\mathrm{T}}(\omega)\mathbf{u}(\omega)$ and $\mathbf{q}(\omega) = \mathbf{H}(\omega)\mathbf{z}(\omega)$.

Our goal is to send a specific wave from the virtual array and capture the corresponding echoes received at the virtual array. However, since we cannot directly excite the virtual transducers, we use the real transducers to transmit a wave that, after propagating through the layer, generates the desired transmit field. We first assign values to $\mathbf{p}(\omega)$ and then proceed to estimate the input $\mathbf{u}(\omega)$ required to have that field. Once we have determined this input $\mathbf{u}(\omega)$, we obtain the measurements at the real array for this particular input, which we have defined as $\mathbf{q}(\omega)$. However, to complete our task, we require the received echoes at the virtual array $\mathbf{z}(\omega)$. Therefore, we need to estimate $\mathbf{z}(\omega)$ from $\mathbf{q}(\omega)$. These steps are depicted by the blue arrows in Figure 2.

A. Obtaining the virtual array model

The estimation of $\mathbf{z}(\omega)$ involves two inverse problems. As discussed in Section II, these inverse problems can be effectively addressed using a matched filtering approach, even if the medium in between is not homogeneous.

Following (9) we can hence estimate $\mathbf{u}(\omega)$ and $\mathbf{z}(\omega)$ as

$$\hat{\mathbf{u}}(\omega) = \mathbf{H}^{*}(\omega)\mathbf{p}(\omega), \quad \hat{\mathbf{z}}(\omega) = \mathbf{H}^{\mathrm{H}}(\omega)\hat{\mathbf{q}}(\omega). \tag{13}$$

Note however that if more accurate estimates need to be obtained, we can follow an approach similar to the regularized inverse used in (8).

The only step that finally remains to be discussed is the step from $\hat{\mathbf{u}}(\omega)$ to $\hat{\mathbf{q}}(\omega)$, which consists of transmitting the $\hat{\mathbf{u}}(\omega)$ waves from the real array and measuring $\hat{\mathbf{q}}(\omega)$.

This leads to $\hat{\mathbf{q}}(\omega) = \mathbf{Y}(\omega)\hat{\mathbf{u}}(\omega)$, with $\mathbf{Y}(\omega)$ as defined in (12). Combining all steps together, we obtain:

$$\begin{aligned}\hat{\mathbf{z}}(\omega) &= \mathbf{H}^{\mathrm{H}}(\omega)\mathbf{Y}(\omega)\mathbf{H}^{*}(\omega)\mathbf{p}(\omega) \\ &= \mathbf{H}^{\mathrm{H}}(\omega)\mathbf{H}(\omega)\tilde{\mathbf{G}}(\omega)\text{diag}(\mathbf{x})\tilde{\mathbf{G}}^{\mathrm{T}}(\omega)\mathbf{H}^{\mathrm{T}}(\omega)\mathbf{H}^{*}(\omega)\mathbf{p}(\omega).\end{aligned}$$

The pressure at the virtual array when sending $\mathbf{p}(\omega)$ from the virtual array would be $\mathbf{z}(\omega) = \tilde{\mathbf{G}}(\omega)\text{diag}(\mathbf{x})\tilde{\mathbf{G}}^{\mathrm{T}}(\omega)\mathbf{p}(\omega)$. Now we clearly see that $\hat{\mathbf{z}}(\omega) \sim \mathbf{z}(\omega)$ if $\mathbf{H}^{\mathrm{H}}(\omega)\mathbf{H}(\omega) \sim \mathbf{I}$, where $\sim$ means proportional to. In reality this is not exactly the case though. First of all, there might be off-diagonal elements and second, the diagonal elements could be all different.

As we mentioned earlier, the model in (12) is applicable for a Nyquist sampled virtual array. When the real array is also Nyquist sampled, i.e., $M = V$, $\mathbf{H}^{\mathrm{H}}(\omega)\mathbf{H}(\omega)$ is close to a scaled identity. However, when we start undersampling, i.e., $M < V$, we begin to violate this assumption. Still, the coding mask helps us in this respect, since we can expect that a coding mask leads to a smaller focal spot and hence an improved resolution when applying time-reversal on an undersampled array.

B. Virtual transmit schemes

So far, we have considered a single transmission from the virtual array. However, it is possible to derive the received echoes for multiple transmissions. We define $\mathbf{p}_k(\omega)$ as the input for the k-th transmission. From each one, we derive the corresponding $\hat{\mathbf{z}}_k(\omega)$. We can write everything in a single equation if we stack the vectors in two matrices $\hat{\mathbf{Z}}(\omega) = [\hat{\mathbf{z}}_1(\omega) \dots \hat{\mathbf{z}}_k(\omega) \dots, \hat{\mathbf{z}}_K(\omega)]$ and $\mathbf{P}(\omega) = [\mathbf{p}_1(\omega) \dots \mathbf{p}_k(\omega) \dots, \mathbf{p}_K(\omega)]$. We derive $\hat{\mathbf{Z}}(\omega)$ from $\mathbf{P}(\omega)$ as:

$$\hat{\mathbf{Z}}(\omega) = \mathbf{H}^{\mathrm{H}}(\omega)\mathbf{Y}(\omega)\mathbf{H}^{*}(\omega)\mathbf{P}(\omega). \tag{14}$$

Let us now define how to set $\mathbf{P}(\omega)$ for different transmit schemes from the virtual array. In each transmission k, the j-th element of the array sends a pulse that is delayed by $\tau_{j,k}$ and has an amplitude of $a_{j,k}$. We can represent this as $P_{j,k}(\omega) = a_{j,k}e^{-i\omega\tau_{j,k}}$, where $P_{j,k}(\omega)$ is the input for the j-th element during the k-th transmission. The value $a_{j,k}$ is binary and is set to 1 if the j-th element is active during the k-th transmission, 0 otherwise.

As our goal is to apply DAS at the virtual array, we focus on transmission schemes that allow us to use DAS. Specifically, we consider three transmission schemes: SA, single plane wave and plane-wave compounding. We define how to set the coefficients for these three cases:

- For an SA scan, the number of transmissions K is equal to the number of virtual transducers V. The coefficient $a_{j,k}$ is 1 for $j = k$, 0 otherwise. The delays $\tau_{j,k}$ are set to zero. Then, $\mathbf{P}(\omega) = \mathbf{I}$ for all ω.
- For a single plane wave transmission, $K = 1$, $a_j = 1$ and $\tau_j = 0$ for all j. Then, $\mathbf{P}(\omega) = \mathbf{1}$ for all ω.
- In the case of plane-wave compounding, K waves at K angles $\theta_1, ..., \theta_K$ are sent. The amplitude coefficients are $a_{j,k} = 1$ for all j, k. The delays can be easily computed based on the array pitch, the speed of sound, and the desired angle.

C. DAS at the virtual array

In this section, we present the proposed image formation method, outlining all its steps. After selecting the transmit scheme and computing the corresponding $\mathbf{P}(\omega)$, the image is computed as follows:

1) Compute $\hat{\mathbf{Z}}(\omega) = \mathbf{H}^{\mathrm{H}}(\omega)\mathbf{Y}(\omega)\mathbf{H}^{*}(\omega)\mathbf{P}(\omega)$
2) Perform an inverse Fourier-transform on each $\hat{Z}_{j,k}(\omega)$
3) Compute each x_n as $\hat{x}_n = \sum_j \sum_k \hat{z}_{j,k}(\delta_{k,n} + \tilde{\tau}_{j,n})$

The delay $\tilde{\tau}_{j,n}$ represents the propagation time from point j on the virtual array to pixel n, while $\delta_{k,n}$ is the time it takes for the k-th transmit field to reach pixel n. It is different for each kth transmission, as we send different types of waves.

V. Simulation Results

A. Simulation settings

We tested our method on simulated data acquired with the k-Wave toolbox [9]. We consider a grid of 1024 (depth) $\times$512 (width) grid points with a isotropic spacing of 90 μm and a sampling frequency of $f_s = 57$ MHz.

We will consider two arrays: one fully sampled and one undersampled. For the fully sampled array we considered an aperture size of $M = 80$ transceivers with a pitch of 270 μm operating at $f_0 = 2.8$ MHz with a $B = 6$ MHz bandwidth. These parameters mimic a clinical phased array used for transcranial and cardiac imaging. For the undersampled array we simply average 8 transceivers signals into 1, i.e., $M = 10$.

A plastic coding mask was simulated using adjacent pillars of different heights with a speed-of-sound of 2750 m/s and a density of 940 kg/m^3. The pillar width is 1 mm and the thickness of the pillars varies randomly in the interval [2,8] mm. The virtual array is at 10 mm depth and it has the same aperture and pitch of the fully sampled real array.

The imaging phantom consists of scattering points (speed-of-sound = 2050 m/s and density = 500 kg/m^3) within a homogeneous medium (speed-of-sound = 1540 m/s and density = 997 kg/m^3). For the first result we used one scattering point and for the second result we used a grid of points.

For each imaging scenario we simulated two SA scans: one without and one with scattering points. Subtracting the first simulated transceiver data from the second simulation yields clean signals originating only from the scattering points and without, e.g., unwanted mask reflections. In the simulation without scattering points, additional sensors are placed at a region-of-interest (ROI) (red rectangle in Fig. 4(A)). This sensor data was used to make the matrices $\mathbf{G}(\omega_1), .., \mathbf{G}(\omega_L)$ and to generate the matrix $\mathbf{A}$ as in (4) in order to compare the new DAS-based results with the full model-based results. Moreover, zero-mean white Gaussian noise is added to this data yielding an SNR of 20 dB.

B. Result for single point phantom

We consider the data for the single point simulation and we derive the echoes at the virtual array for a plane wave transmission. For this transmission, the received echoes form a parabola in the space-time domain. Our goal is to retrieve this parabolic shape with our method.

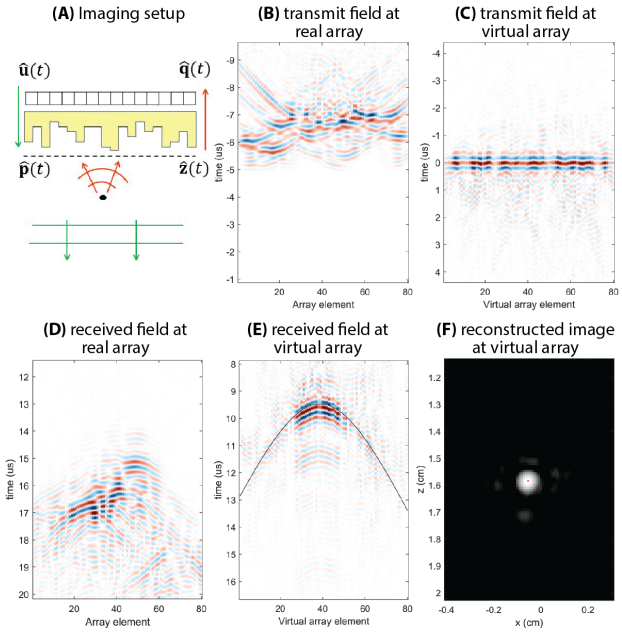

Fig. 3. Transmitting a plane wave through the coding mask (A) is possible by transmitting the waveforms in (B), which will form a plane wave at the virtual array (C). From the received echoes at the real array (D), we derive the echoes at the virtual array (E). The data in (E) is used to reconstruct the image in (F) using DAS

First, we derive the input needed to obtain a plane wave transmission on the virtual array as $\hat{\mathbf{u}}(\omega) = \mathbf{H}^{*}(\omega)\mathbf{1}$. The time domain representation is shown in Figure 3(B). The transmit field at the virtual array is $\hat{\mathbf{p}}(\omega) = \mathbf{H}^{\mathrm{T}}(\omega)\mathbf{H}^{*}(\omega)\mathbf{1}$. The time domain is shown in Figure 3(C).

The echoes at the real array when sending $\hat{\mathbf{u}}(\omega)$ are computed as $\hat{\mathbf{q}}(\omega) = \mathbf{Y}(\omega)\hat{\mathbf{u}}(\omega)$. The time domain representation is shown in Figure 3(D). Here, we see aberrations on a wavefront which should be parabolic considering a point source in a homogeneous medium. To make them parabolic, we compute $\hat{\mathbf{z}}(\omega) = \mathbf{H}^{\mathrm{H}}(\omega)\mathbf{q}(\omega)$, shown in Figure 3(E).

Finally, an image is reconstructed using the data $\hat{\mathbf{z}}(\omega)$ and a DAS implementation. The image envelope signal is computed using a Hilbert transform and log compressed for visualisation purposes.

C. Results for grid phantom

For the grid phantom, we now compare the proposed method with the full model-based matched filter as shown in (9). We generate the model matrix $\mathbf{A}$ for the points within the ROI and apply matched filtering on the array data. The resulting image is displayed in Figure 4(B). Our method shown in Figure 4(C) performs similarly.

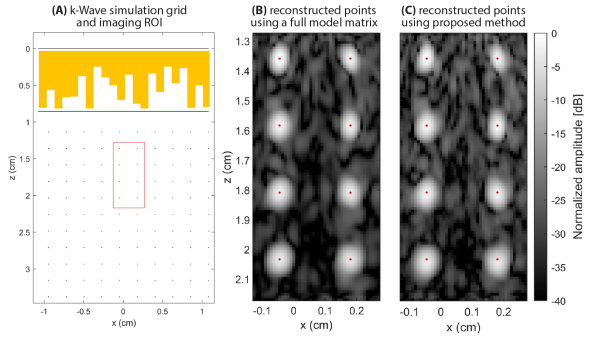

Fig. 4. (A) Set-up for k-Wave simulations with a real array (black line), coding mask (yellow) and virtual array (blue line) and ROI (red rectangle) used for comparing matched filter reconstruction with model matrix (B) and the proposed method (C).

Finally, we apply the method on the undersampled array data. From the data measured at an array consisting of $M = 10$ transceivers, we derive the measurements at a virtual array consisting of $V = 80$ elements. The imaging resulting from applying DAS on the measurements is shown in Figure 5. Clearly, the mask is crucial here.

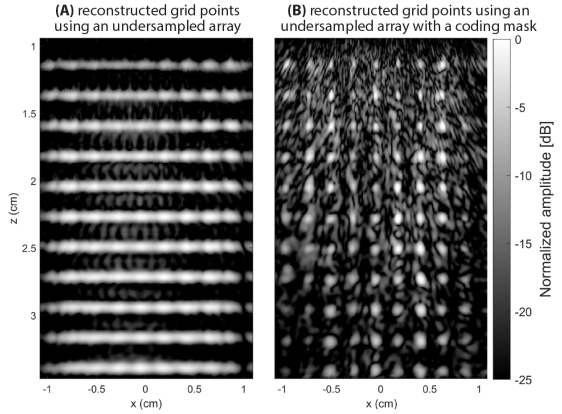

Fig. 5. Grid point reconstruction using a spatially undersampled array (a) without coding mask and (b) with coding mask.

VI. Discussion and Conclusion

Our study presents an alternative solution to the image reconstruction challenge in computational ultrasound imaging (cUSi), moving away from the large model matrix towards a virtual array concept behind an aberrating medium. This method employs transfer functions to reformat data for compatibility with traditional reconstruction methods such as DAS, which are fast and have small memory requirements.

We established the mathematical validity of this method and demonstrated its effectiveness with imaging results. Figure 3 highlights the processing steps of the proposed method with a virtual plane wave transmission and the reception of virtual hyperbolas which were not perfect possibly due to distortive effects like attenuation and refraction by the mask.

Figure 4 underscored that even a basic DAS implementation could closely emulate full matrix reconstruction results, minimizing the need for extensive model matrices. Despite the reduced size of matrix $\mathbf{H}$, its practical implementation on hardware like GPUs needs more exploration. Moreover, the relationship between the number of virtual transmissions and image quality is another area for future study. Figure 5 shows that in the case of a spatially undersampled array, an aberrating coding mask is needed to regain spatial focusing at the expense of SNR. While we focused on a virtual array behind an aberrating layer, this method could apply to any encoding medium, provided the imaging medium is homogeneous.

In essence, our paper provides a method that may simplify cUSi implementation by replacing large storage-demanding model matrices with a virtual array approach. This strategy opens up the possibility of more efficient hardware utilization, particularly in contexts where unknown aberrators, such as the human skull, complicate imaging. Our approach could promote broader adoption of cUSi and serve as a valuable tool for future research in imaging through all kinds of unknown aberrators.

References

[1] A. Derode, A. Tourin, and M. Fink, "Ultrasonic pulse compression with one-bit time reversal through multiple scattering," *Journal of applied physics*, vol. 85, no. 9, pp. 6343–6352, 1999.

[2] G. Montaldo, D. Palacio, M. Tanter, and M. Fink, "Time reversal kaleidoscope: A smart transducer for three-dimensional ultrasonic imaging," *Applied physics letters*, vol. 84, no. 19, pp. 3879–3881, 2004.

[3] P. Kruizinga, P. van der Meulen, A. Fedjajevs, F. Mastik, G. Springeling, N. de Jong, J. G. Bosch, and G. Leus, "Compressive 3d ultrasound imaging using a single sensor," *Science advances*, vol. 3, no. 12, p. e1701423, 2017.

[4] J. Janjic, P. Kruizinga, P. Van Der Meulen, G. Springeling, F. Mastik, G. Leus, J. G. Bosch, A. F. van der Steen, and G. van Soest, "Structured ultrasound microscopy," *Applied Physics Letters*, vol. 112, no. 25, 2018.

[5] M. D. Brown, B. S. Generowicz, S. Dijkhuizen, S. K. Koekkoek, C. Strydis, J. G. Bosch, P. Arvanitis, G. Springeling, G. J. Leus, C. I. De Zeeuw *et al.*, "Four-dimensional computational ultrasound imaging of brain hemodynamics," *Science advances*, vol. 10, no. 3, p. eadk7957, 2024.

[6] P. van der Meulen, P. Kruizinga, J. G. Bosch, and G. Leus, "Coding mask design for single sensor ultrasound imaging," *IEEE Transactions on Computational Imaging*, vol. 6, pp. 358–373, 2019.

[7] V. Perrot, M. Polichetti, F. Varray, and D. Garcia, "So you think you can das? a viewpoint on delay-and-sum beamforming," *Ultrasonics*, vol. 111, p. 106309, 2021.

[8] M. Tanter, J.-L. Thomas, and M. Fink, "Time reversal and the inverse filter," *The Journal of the Acoustical Society of America*, vol. 108, no. 1, pp. 223–234, 2000.

[9] B. E. Treeby and B. T. Cox, "k-wave: Matlab toolbox for the simulation and reconstruction of photoacoustic wave fields," *Journal of biomedical optics*, vol. 15, no. 2, pp. 021 314–021 314, 2010.