Spatially Variant Ultrasound Image Restoration with Product Convolution

Arthur Floquet, Emmanuel Soubies, Duong-Hung Pham, Denis Kouamé

Spatially Variant Ultrasound Image Restoration with Product Convolution

Arthur Floquet, Emmanuel Soubies, Duong-Hung Pham, Denis Kouame

***Abstract*—The process of ultrasound image formation can generally be modeled, using a linear and shift-invariance approximation, as a convolution. In practice, the point spread function (PSF) is shift-variant. Here, we consider the restoration problem using a shift-variant PSF, where it is modelled as product-convolution. We argue that the ultrasound PSF varies smoothly enough for product-convolution to serve as an efficient and effective direct model for ultrasound image restoration. We present a strategy for constructing the product-convolution operator, and derive an efficient optimization scheme. We finally validate our approach on both simulated and real data, demonstrating state-of-the-art results, while achieving significantly faster processing times.**

***Index Terms*—Deblurring, Product-convolution, Signal processing, Ultrasound imaging**

I. Introduction

Ultrasound (US) imaging is a widely used medical imaging modality, valued for its real-time capability, non-invasiveness, portability, and relatively low cost. The imaging process involves transmitting an US wave into soft body tissue and receiving the backscattered signal, known as the Radio-Frequency (RF) signal. This RF signal is related to the tissue reflectivity function (TRF), which characterizes the spatial distribution of acoustic impedance variations caused by changes in tissue density and sound speed. A beamforming step is then applied to the RF signal to generate the beamformed RF image. Although this beamformed RF image contains rich details, it is not visually interpretable. Therefore, the standard practice is to generate a B-mode image, a grayscale representation commonly used in clinical settings for its visual interpretability. This is achieved by computing the envelope of the beamformed RF image followed by log compression to obtain the desired dynamic range.

Despite its advantages, US imaging has inherent limitations. One key challenge, in many settings, is its low signal-to-noise ratio (SNR), primarily caused by speckle noise resulting from the interference between backscattered waves. Additionally, there is an inherent trade-off between penetration depth and spatial resolution, largely driven by the relationship between US frequency and the interaction of sound waves with biological tissues. Moreover, the beamforming process can introduce artifacts, as it depends on physical parameters—such as sound speed—that cannot be precisely determined in many complex real-world conditions [1].

This work was supported by the CIMI Excellence Laboratory, ANR grant ANR-11-LABX-0040 within the French State Programme "Investissements d'Avenir"

A. Floquet, E. Soubies, D.-H. Pham, and D. Kouamé are with the IRIT Laboratory, Université de Toulouse, and CNRS, Toulouse 31400, France. Email: firstname.lastname@irit.fr. (*Corresponding author: Arthur Floquet*)

For these reasons, the restoration of US imaging is an active field of research. Image restoration involves estimating the image $\mathbf{x}$ that best explain the observed image $\mathbf{y}$. In the context of US imaging, $\mathbf{y}$ is the beamformed RF image, while $\mathbf{x}$ represents the TRF. To recover $\mathbf{x}$ from $\mathbf{y}$, one first needs to define the image formation operator $\mathcal{H}$, also known as the *direct model*, such that $\mathbf{y} = \mathcal{N}(\mathcal{H}(\mathbf{x}))$, with $\mathcal{N}$ a noising operator. Although US image formation is inherently nonlinear, it is often sufficiently well-approximated by a linear model, leading to $\mathcal{H}$ being modeled as a blurring operation with a shift-variant point spread function (PSF) [2].

In most cases, the shift-variant nature of the PSF is neglected, reducing the image restoration problem to a standard deconvolution [3]–[6]. Although efficient methods for approximating shift-variant blur have been developed [7], [8], only a few recent studies have specifically addressed shift-variant blur in US image reconstruction. Michailovich *et al.* [9] proposed to split the image $\mathbf{y}$ into blocks small enough that the PSF can be considered stationnary on each block. Each block is then deconvolved independently, using a block-estimated PSF. Florea *et al.* [10] assumed only axial variation of the PSF. They used a parametric PSF model that vary for each row of the image, and defined an axially varying blurring operator. In their subsequent work [11], they utilized Field II [12] to simulate a set of PSF at different depths. These PSFs are linearly interpolated to compute the PSF for each row, and the same axially varying blurring operator is employed as the direct model. Finally, Besson *et al.* [13] combined a US propagation operator with a Delay-and-Sum (DAS) operator to create their direct model, which is then inverted using an efficient matrix-free implementation.

Despite the progress made in the aforementioned research, US image restoration that accounts for spatially varying PSF remains an open challenge. In this paper, our contribution are four-fold.

1) We propose the use of product-convolution, a fast approximation of spatially variant blur, as direct model to reconstruct US images. To the best of our knowledge, this approach has not been explored previously in US imaging.
2) We present how to construct the product-convolution operators for US imaging.
3) We propose an efficient solution to the inverse problem using the Alternative Direction Method of Multipliers (ADMM).
4) We validate our approach through experiments on both simulated and real US data.

The remainder of the paper is structured as follows. We start by describing the US image formation process. Then, we explain how this process can be effectively approximated using product-convolution and present a method for constructing product-convolution operators for US imaging. Afterward, we introduce the variational formulation used to recover estimated TRF and derive an efficient optimization strategy to minimize it. Finally, we show our experimental results on both simulated and real datasets to demonstrate that our proposed method achieves state-of-the-art performance, while being faster.

Notations: Vectors are denoted as bold lowercase letters (e.g., $\mathbf{x}$), while matrices are represented by bold uppercase letters (e.g., $\mathbf{H}$). The conjugate-transpose of a matrix or vector is written as $\mathbf{x}^T$ and $\mathbf{H}^T$, respectively. The nth component of a vector $\mathbf{x} \in \mathbb{R}^N$ is x_n, while the (m, n) component of a matrix $\mathbf{H} \in \mathbb{R}^{M\times N}$ is $[\mathbf{H}]_{mn}$. The identity matrix of size N will be denoted by $\mathbf{I}_N$. The discrete Fourier transform (DFT) matrix of size N will be denoted by $\mathbf{F}_N$. The symbols $*$, $\otimes$ and $\odot$ stand for convolution, Kronecker product, and element-wise (Hadamard) product, respectively. Finally, $\mathbf{x}^{\circ 2}$ denotes the element-wise square operation of the vector $\mathbf{x}$.

II. Method

A. US Image Formation Model

An US probe consists of an array of piezoelectric elements that act as electromechanical transducers, converting electrical signals into mechanical vibrations and vice versa. When stimulated by an electrical impulse, these elements emit US waves that propagate through the medium. When the wave encounters an inhomogeneity, known as a scatterer, it reflects back toward the probe. Each probe element detects these echoes over time, forming what is called a Radio Frequency (RF) line. The collection of all RF lines makes up the RF signal. For a single scatterer, this RF signal forms a parabola because the probe elements are positioned at different distances from the scatterer, causing them to receive echoes at slightly different times. To reconstruct a single point from this parabolic signal, a process called beamforming is applied. In biological tissue, scatterers are not isolated, meaning US waves reflect off multiple scatterers before returning to the probe. However, under the assumption of weak scattering—where only a small portion of the wave energy is reflected at each scatterer—these multiple scattering effects becomes negligible, as is often the case in soft tissues. Known as Born's approximation, this assumption simplifies the image formation process.

As explained in [2], ignoring multiple scattering, 2D US image formation can be approximated as a spatially variant blur as described by the equation:

$$y(\mathbf{r}) = \int_{\mathbb{R}^2} \kappa(\mathbf{r} - \mathbf{s}, \mathbf{s}) x(\mathbf{s})\, \mathrm{d}\mathbf{s}, \tag{1}$$

where x is the continuous TRF, y the observed beamformed RF image, and $\mathbf{r} \in \mathbb{R}^2, \mathbf{s} \in \mathbb{R}^2$ are coordinates vector composed of the axial and lateral positions. The operator kernel κ is then interpreted as the space-varying PSF.

The PSF variations are caused by different factors. The first factor is US wave attenuation, which leads to a gradual decay in the PSF amplitude and a phase shift, particularly along the axial direction. The second factor is the type of wave emission. For instance, a focused wave causes the PSF amplitude to increase with depth until it reaches the focal point, after which it begins to decreases. In contrast, a diverging wave leads to larger variations in the PSF along the azimuthal axis when compared to a plane wave. The elevational focus also induce PSF variations, as more out-of-plan signal is received from the out of elevational focus depths.

B. Direct Model: Product-Convolution

In this section, we describe the direct model employed to deblur US images. We begin by outlining the model at the continuous level, followed by a formal definition of the corresponding discrete model.

1) Continuous Model: The numerical evaluation of the integral (1) is computationally expensive. Thus, a number of techniques have been developed to efficiently approximate it, as summarized in [7]. In this work, we approximate (1) using product-convolution, a method initially introduced in [15]. Product-convolution models the varying blur as the sum of K weighted convolutions with kernels $h_k : \mathbb{R}^2 \to \mathbb{R}$ and weights maps $w_k : \mathbb{R}^2 \to \mathbb{R}$, $k = 1, \ldots, K$. This approximation is given by the following equation:

$$y(\mathbf{r}) \approx \left(\sum_{k=1}^{K} h_k * (w_k \cdot x) \right)(\mathbf{r}). \tag{2}$$

The main computational complexity of the product-convolution resides in the K convolution operations. Thus, the fewer kernels and weight-maps are needed to accurately represent the integral (1), the better. Escande and Weiss [8] showed that when the space-varying PSF $\kappa(\cdot, \mathbf{s})$ varies slowly with respect to the space variable $\mathbf{s}$, (1) can be well approximated by (2) with a small value of K.

We argue that this condition holds in US imaging, since the factors causing the PSF variation do not induce discontinuities. However, the actual construction of the families $(h_k)_{k=1}^K$ and $(w_k)_{k=1}^K$ is not trivial. We address this point in Section II-C.

2) Discrete Model: In discrete form, the product-convolution model (2) is expressed as a combination of diagonal and discrete convolution operators. Let $\mathbf{x} \in \mathbb{R}^N$ denote the discrete TRF and $\mathbf{y} \in \mathbb{R}^N$ represent the discrete beamformed RF image, both of which are sampled from x and y on a regular grid. Additionally, let $\mathbf{h}_k \in \mathbb{R}^N$ be the discrete version of h_k and $\mathbf{w}_k \in \mathbb{R}^N$ the discrete version of w_k. All vectors here are arranged in the lexicographical order. From the matrices $\mathbf{W}_k = \mathbf{diag}(\mathbf{w}_k) \in \mathbb{R}^{N\times N}$, we form:

$$\mathbf{W} = (\mathbf{W}_1, \cdots, \mathbf{W}_K)^T \in \mathbb{R}^{KN\times N},$$

such that the product $\mathbf{W}\mathbf{x}$ performs an element-wise multiplication of $\mathbf{x}$ with each of the weight maps $\mathbf{w}_k$. We also define the matrices $\mathbf{H}_k \in \mathbb{R}^{N\times N}$ as circulant matrices formed from the discrete kernels $\mathbf{h}_k$, so that $\mathbf{H}_k\mathbf{x}$ corresponds to the convolution between $\mathbf{h}_k$ and $\mathbf{x}$ with periodic boundary conditions. These matrices $\mathbf{H}_k$ are diagonalizable in the Fourier domain, specifically:

$$\mathbf{H}_k = \mathbf{F}_N^T \mathbf{\Lambda}_k \mathbf{F}_N, \text{ with } \mathbf{\Lambda}_k = \mathbf{diag}(\mathbf{F}_N \mathbf{h}_k),$$

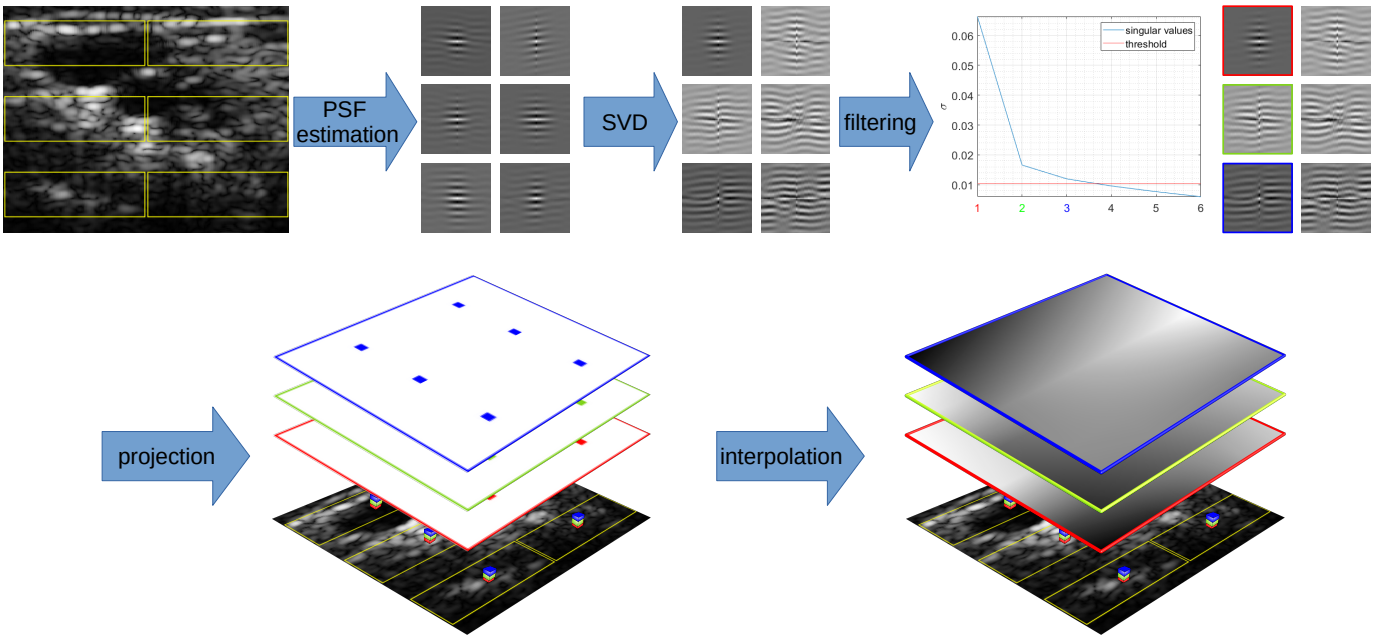

Fig. 1: Construction of the product-convolution kernels and weights, following [14]: First, $P = 6$ PSF $(\boldsymbol{\kappa}_p)_{p=1}^P$ are estimated from an US beamformed image. Then, SVD is performed to extract the P basis vector of the PSF space,$(\mathbf{h}_p)_{p=1}^P$. With singular value filtering, only the first $K = 3$ are retained. Each PSF is projected onto this K-dimensionnal basis, giving the first values of $(\mathbf{w}_k)_{k=1}^K$. Projection coefficients are finally interpolated to complete the weights maps $(\mathbf{w}_k)_{k=1}^K$.

where $\mathbf{F}_N \in \mathbb{C}^N$ denotes the DFT. We then define the following matrix

$$\mathbf{H} = (\mathbf{H}_1, \cdots, \mathbf{H}_K) \in \mathbb{R}^{N \times KN},$$

such that $\mathbf{Hu}$ for $\mathbf{u} = (\mathbf{u}_1, \cdots, \mathbf{u}_K)^T$ sums the the K convolutions $\mathbf{H}_k \mathbf{u}_k$. Note that $\mathbf{H} = \mathbf{F}_N^T \boldsymbol{\Lambda} \mathbf{F}$, with $\boldsymbol{\Lambda} = (\boldsymbol{\Lambda}_1, \cdots, \boldsymbol{\Lambda}_K) \in \mathbb{R}^{N \times KN}$, and

$$\mathbf{F} = \mathbf{I}_K \otimes \mathbf{F}_N = \begin{pmatrix} \mathbf{F}_N & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_N & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{F}_N \end{pmatrix} \in \mathbb{C}^{KN \times KN}.$$

Combining all the aforementioned elements, the discrete form of (2) can be expressed as:

$$\mathbf{y} = \mathbf{HWx} + \mathbf{n}, \tag{3}$$

where $\mathbf{n}$ represents a perturbation term that accounts for noise, approximation errors, and discretization errors.

Leveraging the Fast Fourier Transform (FFT) to efficiently perform convolutions in the Fourier domain, the computational complexity of this model is of order $\mathcal{O}((K+1)N \log_2(N))$. This corresponds to $K+1$ FFTs, each with a complexity of order $\mathcal{O}(N \log_2(N))$ [16].

C. Operators Construction

In this section, we present our strategy to construct the product-convolution operators. The whole pipeline, which follows [14], is illustrated in Figure 1. Specifically, the kernels $\mathbf{h}_k$ and weights $\mathbf{w}_k$ are built as follows.

1) Sample a set of P estimated local PSF, $\boldsymbol{\kappa}_p \in \mathbb{R}^M$, $p \in \{1, \dots, P\}$, $M < N$, on patches centered at P different position (vectorized pixels) of the input data $\mathbf{y}$, which we denote $i_p \in \{1, \dots, N\}$, forming a regular grid. Note that if prior PSF variations related information is available, the sampling strategy can be adapted to the data. The considered PSF estimation method is presented in Section II-D.
2) Form the matrix $\mathbf{P} = (\boldsymbol{\kappa}_1, \dots, \boldsymbol{\kappa}_P) \in \mathbb{R}^{M \times P}$. Compute its compact Singular Value Decomposition (SVD), $\mathbf{P} = \mathbf{U\Sigma V}^T$. The column vectors of $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_P) \in \mathbb{R}^{M \times P}$ form an orthogonal basis of the PSF subspace (of dimension $P < M$ in $\mathbb{R}^M$). We use those basis vectors as kernel for our product-convolution model, i.e. $\mathbf{h}_k = \mathbf{Z}\mathbf{u}_k$ with $\mathbf{Z} \in \mathbb{R}^{N \times M}$ a zero-padding operator.
3) Select the number of kernels to retain, K. As mentioned in Section II-B, a larger number of kernels results in a more accurate but computationally more expensive model, while a smaller number of kernels leads to a less accurate model with low computationnal burden. We propose a strategy based on singular value filtering, which we detail in Section II-E.
4) Project each estimated PSF $\boldsymbol{\kappa}_p$ onto the subspace formed by $(\mathbf{h}_k)_{k=1}^K$. That is, compute the projection coefficient $\mathbf{C} \in \mathbb{R}^{K \times P}$ defined as:
$$\mathbf{C} = \arg\min_{\tilde{\mathbf{C}}} \left\| (\mathbf{h}_1, \dots, \mathbf{h}_K)\tilde{\mathbf{C}} - \mathbf{P} \right\|_2^2. \tag{4}$$
Since $(\mathbf{h}_k)_{k=1}^K$ forms an orthognal basis, solving (4) gives $[\mathbf{C}]_{kp} = \mathbf{h}_k^T \boldsymbol{\kappa}_p / ||\mathbf{h}_k||_2^2$.
5) The values of the weights $\mathbf{w}_k$ correspond to the projection coefficients. But they are computed explicitly only at points where the PSF are estimated. That is, we have $[\mathbf{w}_k]_{i_p} = [\mathbf{C}]_{kp}$. Thus, we need to interpolate the value

of $\mathbf{w}_k$ on the rest of the domain. Among the different possible interpolation methods, we use natural neighbor interpolation [17], as it produces a smooth function and aligns with the assumption that κ varies smoothly.

D. PSF Estimation

To estimate PSFs, we use the approach proposed by Michailovich *et al.* [9]. Considering a convolution model $\mathbf{y} = \mathbf{h} * \mathbf{x}$, they first compute the Fourier transform of $\mathbf{y}$, $\tilde{\mathbf{y}} = \tilde{\mathbf{h}} \odot \tilde{\mathbf{x}}$, with $\tilde{\cdot}$ denoting the Fourier transform. Then, the complex logarithmic transform is performed, giving:

$$\log|\tilde{\mathbf{y}}| = \log|\tilde{\mathbf{h}}| + \log|\tilde{\mathbf{x}}| \tag{5}$$

$$\measuredangle\tilde{\mathbf{y}} = \measuredangle\tilde{\mathbf{h}} + \measuredangle\tilde{\mathbf{x}}, \tag{6}$$

with $\measuredangle(\cdot)$ being the operator that extract the phase. Then, the PSF estimation problem becomes a filtering problem. The amplitude of $\mathbf{h}$ is found by filtering $\log|\tilde{\mathbf{h}}|$ from $\log|\tilde{\mathbf{x}}|$. This filtering is possible because of the differences in smoothness properties between $\log|\tilde{\mathbf{h}}|$ and $\log|\tilde{\mathbf{x}}|$. Specifically, $\log|\tilde{\mathbf{h}}|$ is assumed to be a much smoother function than $\log|\tilde{\mathbf{x}}|$. Thus, denoising $\log|\tilde{\mathbf{y}}|$ eliminates its $\log|\tilde{\mathbf{x}}|$ component, while preserving $\log|\tilde{\mathbf{h}}|$. The same reasoning can be applied to (6), to compute the phase. However, it is less straightforward because the phase $\measuredangle\tilde{\mathbf{y}}$ is wrapped (i.e., "modulus 2π"), so it contains discontinuities. We refer the reader to [9] for further explanations. To account for variability of the PSF, we perform P PSF estimations, on patches of the US image, with the center of the patches forming the regular grid $\mathcal{P}$.

E. Number of Product-Convolution Kernels

As discussed in Section II-B, selecting the number of kernels K in the product-convolution model is a crucial decision, as it determines the balance between computational efficiency and accuracy. We argue that, for US imaging, a relatively small number of kernels are sufficient to represent the varying PSF. To validate this, we conduct the following experiment.

We define $\mathcal{PC}_K$ the product-convolution model using K kernels. Given P estimated PSFs, the best possible model is $\mathcal{PC}_P$. Ideally, we want to keep K as low as possible, while maintaining the same *relative* performance compared to $\mathcal{PC}_P$. To assess this relative performance, we generate a synthetic TRF $\mathbf{x}$, then simulate 300 PSFs (arranged on a grid) using SIMUS [18], and create the "full" model, $\mathcal{PC}_{300}$.The US image is then simulated using this model:

$$\mathbf{y}_{300} = \mathcal{PC}_{300}(\mathbf{x}) + \mathbf{n},$$

with $\mathbf{n}$ Gaussian noise such that $\mathbf{y}_{300}$ has a Signal-to-Noise Ratio (SNR) of 25dB.

Then, we compute an estimated TRF $\hat{\mathbf{x}}_{300}$ using the same model, $\mathcal{PC}_{300}$. We then instantiate models with fewer kernels, $\mathcal{PC}_{K<300}$, and use them to reconstruct the corresponding estimated TRFs $\hat{\mathbf{x}}_{K<300}$. The relative performance is measured using $\mathrm{PSNR}(\hat{\mathbf{x}}_{K<300}, \hat{\mathbf{x}}_{300})$.

The results, shown in Figure 2, indicate that the singular values of $\mathbf{P}$ quickly decrease, and the relative performance of $\mathcal{PC}_K$ improves correspondingly before stabilizing. This indicates that beyond a certain number of kernels, adding more kernels does not significantly enhance performance. Based on this experiment, we set a threshold of $0.06\sigma_1$, where σ_1 is the first and largest singular value, and keep only the kernels $\mathbf{h}_k$ with related singular value above this threshold.

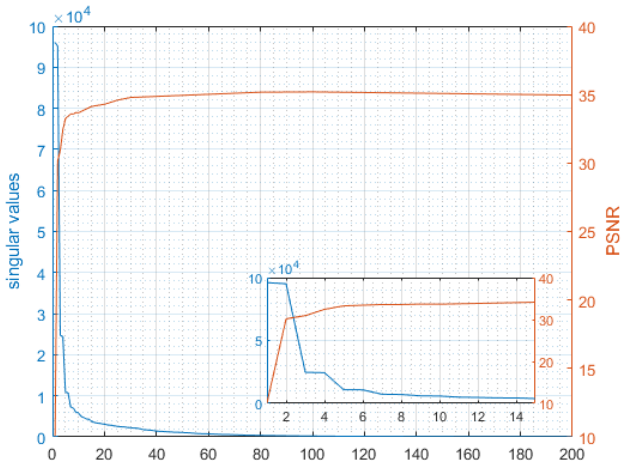

Fig. 2: Singular values against relative performance. Inset graph is a zoom limited to the first 15 singular values.

F. Optimization

In this section, we set the variational formulation used to solve the inverse problem, i.e. to recover an estimated TRF. We then describe ADMM, the optimization algorithm used to solve it. Finally, we present a splitting allowing for an efficient implementation of ADMM.

We reconstruct an estimated TRF $\hat{\mathbf{x}}$ by solving the optimization problem:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{R}^N} \frac{1}{2}||\mathbf{HWx} - \mathbf{y}||_2^2 + \lambda\mathcal{R}(\mathbf{x}), \tag{7}$$

with $\mathcal{R}$ a regularization term and λ the associated regularization strength parameter. A recent practice in US imaging is to use $\ell_p; 0 < p \leq 2$ norm as regularization [13], [19]. In the case $p \neq 2$, ℓ_p is non-differentiable. Thus, we propose to tackle the minimization problem with ADMM.

ADMM [20] is an optimization algorithm that minimizes problems of the form:

$$\min_{\mathbf{x}} \frac{1}{2}||\mathbf{Ax} - \mathbf{b}||_2^2 + \sum_{q=1}^{Q} f_q(\mathbf{B}_q\mathbf{x}), \tag{8}$$

where f_q are convex functions. It uses variable splitting, i.e. the introduction of auxiliary variables $\mathbf{u}_q = \mathbf{B}_q\mathbf{x}$, to decouple the different operators and functions involved. This results in an equivalent constrained optimization problem:

$$\min_{\mathbf{x},(\mathbf{u}_q)_{q=1}^Q} \frac{1}{2}||\mathbf{Ax} - \mathbf{b}||_2^2 + \sum_{q=1}^{Q} f_q(\mathbf{u}_q), \text{ s.t.} \forall q, \mathbf{u}_q = \mathbf{B}_q\mathbf{x}. \tag{9}$$

To get back to an unconstrained optimization problem, the augmented Lagrangian $\mathcal{L}$ is used. It transforms constraints into

penalties. The problem is finally casted as the search for a saddle point of $\mathcal{L}$:

$$\min_{\mathbf{x},(\mathbf{u}_q)_{q=1}^Q} \max_{(\mathbf{v})_{q=1}^Q} \mathcal{L}\left(\mathbf{x},(\mathbf{u}_q)_{q=1}^Q,(\mathbf{v}_q)_{q=1}^Q\right), \tag{10}$$

$$\text{with } \mathcal{L}\left(\mathbf{x},(\mathbf{u}_q)_{q=1}^Q,(\mathbf{v}_q)_{q=1}^Q\right) = \frac{1}{2}||\mathbf{A}\mathbf{x}-\mathbf{b}||_2^2 + \sum_{q=1}^{Q}\left(f_q(\mathbf{u}_q) + \frac{\rho_q}{2}\left\|\mathbf{u}_q - \mathbf{B}_q\mathbf{x} - \frac{\mathbf{v}_q}{\rho_q}\right\|_2^2 - \frac{1}{2\rho_q}||\mathbf{v}_q||_2^2\right) \tag{11}$$

where $\rho_q > 0$ are parameters that influence convergence speed.This problem is tackled through alternating minimization and one ADMM iteration is thus made of three steps:

1) Auxiliary variables update: for each q, minimize $\mathcal{L}$ with respect to the auxiliary variable $\mathbf{u}_q$,

$$\begin{aligned}\mathbf{u}_q^{k+1} &= \arg\min_{\mathbf{u}} \frac{\rho_q}{2}\|\mathbf{u}-\mathbf{B}_q\mathbf{x}^k - \mathbf{v}_q^k/\rho_q\|_2^2 + f_q(\mathbf{u}_q) \\ &= \text{prox}_{\frac{1}{\rho_q}f_q}\left(\mathbf{B}_q\mathbf{x}^k + \frac{\mathbf{v}_q^k}{\rho_q}\right).\end{aligned} \tag{12}$$

2) Solution update: update the solution $\mathbf{x}$ by minimizing $\mathcal{L}$ with respect to $\mathbf{x}$,

$$\mathbf{x}^{k+1} = \left(\mathbf{A}^T\mathbf{A} + \sum_{q=1}^{Q}\rho_q\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\mathbf{s}^k, \tag{13}$$

$$\text{with } \mathbf{s}^k = \left(\mathbf{A}^T\mathbf{b} + \sum_{q=1}^{Q}\rho_q\mathbf{B}_q^T\left(\mathbf{u}_q^{k+1} - \frac{\mathbf{v}_q^k}{\rho_q}\right)\right).$$

3) Lagrange multiplier update: for each q, $\mathbf{v}_q$ is updated according to

$$\mathbf{v}_q^{k+1} = \mathbf{v}_q^k + \rho_q\left(\mathbf{B}_q\mathbf{x}^{k+1} - \mathbf{u}_q^{k+1}\right). \tag{14}$$

There are two potential bottlenecks in ADMM: the computation of the proximal operators in the auxiliary variables update, and the matrix inversion in the solution update. If there is no closed form solution for those steps, iterative algorithm, will be used to solve them. This will lead to much slower, and potentially inexact iterations if a limited number of iterations is performed. This can hinder convergence of ADMM. A smart splitting choice can lead to closed form solutions of proximal operators and/or matrix inverse. The most natural splitting strategy would be to only decouple the regularization term. However, this leads to an unefficient solution update, as a subroutine would be needed for matrix inversion.

We propose an alternative splitting strategy that consists in splitting both the regularization term and the data fidelity term. Specifically, we identify (7) within (8) by setting $Q = 2$ and

$$\begin{aligned}&\mathbf{A} = \mathbf{0},\ \mathbf{b} = \mathbf{0} \\ &\mathbf{B}_1 = \mathbf{W},\ f_1(\cdot) = \frac{1}{2}||\mathbf{H}\cdot - \mathbf{y}||_2^2 \\ &\mathbf{B}_2 = \mathbf{I}_N,\ f_2(\cdot) = \lambda\mathcal{R}(\cdot).\end{aligned} \tag{15}$$

With this splitting, the matrix to inverse in step (13) is $(\rho_1\mathbf{W}^T\mathbf{W} + \rho_2\mathbf{I}_N)$. This is a diagonal matrix, since $\mathbf{W}^T\mathbf{W}$ is itself diagonal:

$$\mathbf{W}^T\mathbf{W} = \sum_{k=1}^{K}\mathbf{W}_k^T\mathbf{W}_k = \text{diag}\left(\sum_{k=1}^{K}\mathbf{w_k}^{\circ 2}\right) \in \mathbb{R}^{N\times N}, \tag{16}$$

where $(\cdot)^{\circ 2}$ is the element-wise square operation. Its inversion, involved in (13), is thus a simple element-wise division.

For regularization, we use the ℓ_1 norm or the $\ell_{1.5}$ norm. In both case, a closed form of the proximal operator is known:

$$\left[\text{prox}_{\lambda\ell_1}(\mathbf{x})\right]_i = \text{sgn}(x_i)\max(0, x_i - \lambda) \tag{17}$$

$$\left[\text{prox}_{\lambda\ell_{1.5}}(\mathbf{x})\right]_i = x_i + \frac{9}{8}\lambda^2\text{sgn}(x_i)\left(1 - \sqrt{1+\frac{16|x_i|}{9\lambda^2}}\right) \tag{18}$$

We now show that the the proximal operator of f_1 can also be efficiently computed in the Fourier domain. The proof of the following proposition is given in Appendix A.

Proposition 1: The proximal operator of the function $f_1(\cdot) = \frac{1}{2}\|\mathbf{H}\cdot - \mathbf{y}\|_2^2$ admits a closed form expression,

$$\text{prox}_{\frac{1}{\rho_1}f_1}(\mathbf{x}) = \left(\frac{1}{\rho_1}\mathbf{I}_{KN} - \frac{1}{\rho_1^2}\mathbf{F}^T\mathbf{\Lambda}^T\mathbf{D}^{-1}\mathbf{\Lambda}\mathbf{F}\right)\mathbf{z} \tag{19}$$

where $\mathbf{z} = \mathbf{H}^T\mathbf{y} + \rho_1\mathbf{x}$ and $\mathbf{D} = \mathbf{I}_N + \mathbf{\Lambda}\mathbf{\Lambda}^T$.

With $\mathbf{H}^T\mathbf{y}$ being constant and therefore computed only once, the computational complexity of the proximal step of f_1 is of order $\mathcal{O}(2KN\log_2(N))$. The splitting choice (15) allows for each ADMM step to have an efficient closed form solution. This limits the overall computational cost of one ADMM iteration to $\mathcal{O}(2KN\log_2(N))$. In all reported experiments, we used this double splitting strategy, fully written in Algorithm 1. We implemented this reconstruction process within the GlobalBioIm framework [21] and provide the code in an online repository.[1].

Algorithm 1 ADMM optimization scheme

Require:
Observed data $\mathbf{y}$
Operators $\mathbf{W}, \mathbf{F}, \mathbf{\Lambda}$
Regularization term $\mathcal{R}$
Parameters $\lambda, \rho_1, \rho_2, N_{\text{iter}}$
$(\hat{\mathbf{x}}, \mathbf{u}_1, \mathbf{u}_2, \mathbf{v}_1, \mathbf{v}_2) \leftarrow \mathbf{0}$
for $i \leftarrow 1$ to N_{iter} **do**
  $\mathbf{z} \leftarrow \left(\mathbf{H}^T\mathbf{y} + \rho_1\mathbf{W}\hat{\mathbf{x}} + \mathbf{v}_1\right)$
  $\mathbf{u}_1 \leftarrow \left(\frac{1}{\rho_1}\mathbf{I}_{KN} - \frac{1}{\rho_1^2}\mathbf{F}^T\mathbf{\Lambda}^T\mathbf{D}^{-1}\mathbf{\Lambda}\mathbf{F}\right)\mathbf{z}$
  $\mathbf{u}_2 \leftarrow \text{prox}_{\lambda\mathcal{R}}\left(\rho_2\hat{\mathbf{x}} + \mathbf{v}_2\right)$ ▷ see (17) or (18)
  $\mathbf{w} \leftarrow \left(\mathbf{W}^T(\rho_1\mathbf{u}_1 - \mathbf{v}_1) + \rho_2\mathbf{u}_2 + \mathbf{v}_2\right)$
  $\hat{\mathbf{x}} \leftarrow \left(\rho_1\mathbf{W}^T\mathbf{W} + \rho_2\mathbf{I}_N\right)^{-1}\mathbf{w}$
  $\mathbf{v}_1 \leftarrow \mathbf{v}_1 + \rho_1(\mathbf{W}\hat{\mathbf{x}} - \mathbf{u}_1)$
  $\mathbf{v}_2 \leftarrow \mathbf{v}_2 + \rho_2(\hat{\mathbf{x}} - \mathbf{u}_2)$
end for

[1]All of our code, data and results can be found here

III. Numerical Experiments

A. Datasets

1) In Silico Data: For our simulations experiments, we use SIMUS [18]. We use the linear 64-elements probe 'P4-2v' to transmit a single, not tilted plane wave. The medium is homogeneous, with a sound speed $c = 1540$m/s. It follows a linear frequency dependant attenuation law, with the attenuation parameter set at $\alpha = 1$dB/cm/MHz. Note that even though the simulated data $\mathbf{y}$ is an image, and that the reconstructed TRF $\hat{\mathbf{x}}$ is also an image, there is no ground-truth TRF $\mathbf{x}$, because SIMUS take as input a list of scatterer, and not a TRF. To simulate our data, we first create a synthetic TRF, and assign physical dimension to it. Then we randomly draw scatterers inside the virtual space. Finally, each scatterer $\boldsymbol{\gamma}$ at position $\mathbf{r}$ is assigned as reflectivity x_i, with i being the index matching the physical position $\mathbf{r}$. Our synthetic TRF is of size $50\text{mm} \times 57\text{mm}$, and is composed of three hyperechogenic ellipsoids. Zero-mean Gaussian noise with standard deviation with $\sigma_{\mathbf{n}}$ is added to the raw RF signal, with an input signal-to-noise ratio, defined as i-SNR $= 10\log_{10}(||\mathbf{y}||_2^2/N\sigma_{\mathbf{n}}^2)$, of 25dB.

2) In Vitro Data: To have a more realistic image where structures are still known, we use the data proposed in [22]. It was acquired on the Multi-Purpose Multi-Tissue US Phantom Model 040GSE (CIRS, Norfolk, USA), using a Verasonics T256 R256 system with an L12-5 50mm linear array US transducer operating at a center frequency of 7.8 MHz.

3) In Vivo Data: For our *in vivo* experiments, we use the data proposed by Besson *et al.* [13]. They used the linear-array probe L12-5 50 mm, to acquire measurements of two carotids on a Verasonics Vantage 256 system (Redmond, WA, USA). A single plane wave (5 MHz, 1-cycle excitation) with normal incidence was transmitted, without apodization.

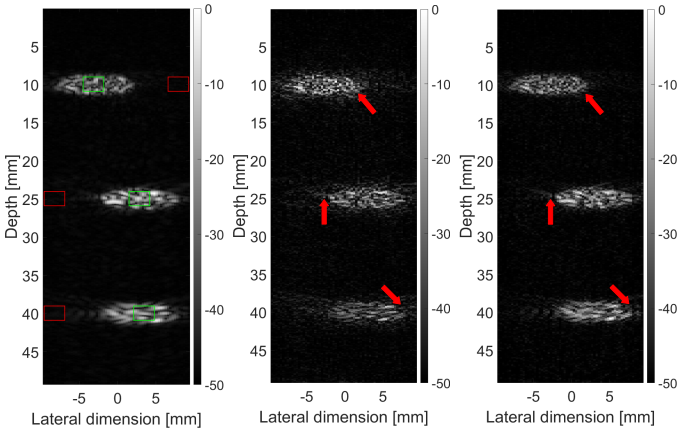

Fig. 3: Restoration with estimated PSF. (a): Input data (rectangle indicates patches $\mathbf{r}_b$ and $\mathbf{r}_t$ on which metrics are computed); (b): Single convolution model; (c): Product-convolution model.

B. Simulation Results

1) Restoration with estimated PSF: In the simulation context, we focus on comparing the reconstruction quality obtained using a convolution model, and our proposed product-convolution model. Figure 3 presents such a comparison. For the convolution model, we estimated a single PSF on the whole image, used ℓ_1 regularization, and set $\lambda = 1\text{e}-4$. For product-convolution, we estimated 20 PSFs (2 in the aziumthal direction times 10 in the axial direction) and kept $K = 5$ basis elements to build the operator (see Section II-C). We also used ℓ_1 regularization, and set $\lambda = 7.5\text{e}-3$.

Ellipsoid Number	Method	CNR	TCR [dB]
1	Product-convolution	**49.5**	**27.7**
	Single convolution	22.3	19.2
2	Product-convolution	**51.5**	**28.8**
	Single convolution	33.7	21.1
3	Product-convolution	30.1	**26.5**
	Single convolution	**33.6**	20.9

TABLE I: Simulation results metrics

In order to quantify the quality of the reconstructions, we use the Contrast-to-Noise Ratio (CNR) and the Tissue-to-Clutter Ratio (TCR). They are defined over two regions of the normalized envelope, $\mathbf{r}_b$ representing the background, and $\mathbf{r}_t$ representing tissue, as:

$$\text{CNR}(\mathbf{r}_b, \mathbf{r}_t) = \frac{|\mu_b - \mu_t|}{\sqrt{\frac{\sigma_b^2+\sigma_t^2}{2}}}, \ \text{TCR} = 20\log_{10}\left(\frac{\mu_t}{\mu_b}\right).$$

where μ_i and σ_i are the mean and standard deviation over $\mathbf{r}_i$. To avoid being impacted by the attenuation, $\mathbf{r}_b$ and $\mathbf{r}_t$ are selected at the same depth, as can be seen in Figure 3.a.

In Figure 3.b, we observe artefacts surrounding the ellipsoids, highlighted by red arrows. These artefacts are precisely what regularization aims to eliminate. However, as the mismatch between the direct model and physical reality (in this case, the simulation software) increases, the artefacts become more pronounced. At a certain point, eliminating them without corrupting the signal becomes impossible. This is why the ellipsoids appear slightly too sparse, yet artefacts persist. In contrast, in Figure 3.c, where product-convolution is used, we can achieve a regularization strength that effectively eliminates or significantly attenuates the artefacts while preserving good signal recovery. These observations are in line with the metrics reported in Table I.

2) Restoration with Simulated PSF: Because of the linearity of convolution, $a\mathbf{h} * \mathbf{x} = \mathbf{h} * a\mathbf{x}$. Thus, the PSF estimation algorithm cannot properly estimate the PSF amplitude from the sole data $\mathbf{y}$ without prior information. To observe the full potential of product-convolution, i.e. when the PSFs are exactly known, we propose the following experiment. As in Section III-B, we set ourselves in a simulation setup, with two differences: We turn off the Time Gain Compensation (TGC), and instead of using PSFs estimated from $\mathbf{y}$, we use 300 simulated PSF, arranged in a 3×100 grid. Each PSF is generated by providing SIMUS with a single scatterer located at the corresponding position on the grid. Since they are simulated, their amplitude are known, and take into account the medium attenuation. In particular, the PSF amplitudes decay with depth. Restoration results are illustrated in Figure 4. We see, as opposed to Figure 3.c, that the ellipsoid at lower depth are enhanced. This is made possible because the decaying

amplitude of PSFs are incorporated in the product-convolution model. If a PSF estimation method could incorporate such information, then leveraging a product-convolution model for restoration might additionally help the TGC.

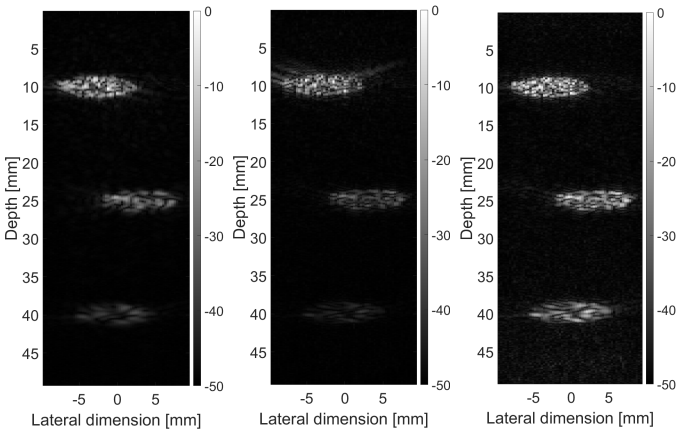

Fig. 4: Restoration with simulated PSF. (a): Input data; (b): Single convolution model; (c): Product-convolution model.

C. In Vitro Results

The phantom data, illustrated in Figure 5, exhibits challenges similar to those found in *in vivo* data. It features sharp structures, here wires, which we aim to deblur as effectively as possible while preserving the visible contrast in other areas, particularly the hyperechoic region at a depth of 28mm. We evaluate the contrast performance using the TCR. We also assess resolution by determining the proportion of pixels exceeding 0.75 in the normalized 2D autocorrelation function of the envelope, denoted as α. We define the resolution gain (RG) of an estimate $\hat{\mathbf{x}}$ as $\alpha(\mathbf{y})/\alpha(\hat{\mathbf{x}})$. We compute the RG on 2.14mm $\times$ 2.14mm patches centered around the wires.

Since no TGC was applied during the acquisition of the phantom data, we evaluated the attenuation using the mean signal intensity at each depth and incorporated this into the estimated PSFs of the product-convolution model. This approach allows the reconstruction to enhance the signal differently at various depths. Consequently, it is possible to significantly deblur the deeper wires, as observed in Figure 5.f and Table II, while maintaining contrast in the hyperechoic region. However, this process also amplifies noise, making it challenging to simultaneously deblur and denoise at depths of 0mm-30mm without compromising the wire signals at 37mm. As a result, the product-convolution method achieves a slightly lower mean resolution gain than the convolution method, as reported in Table II. Specifically, we achieve better resolution gain on the three deeper wires but a lower gain on the four superficial wires.

Additionally, we note that since such phantom images generally contain only a few simple structures, the PSFs tends to exhibit minimal variation, which limits the benefits of considering varying blur.

Method	TCR (dB)	Mean Resolution Gain
Product-convolution	**16.46**	1.82
Single convolution	10.37	**2.13**

TABLE II: Metrics of the phantom reconstructions

Carotid Number	Method	$\mathcal{R}$	λ	N_{iter}
1	Product-convolution	$\ell_{1.5}$	0.0065	93 (ADMM)
	Besson *et al.*	$\ell_{1.5}$	0.005	100 (FISTA)
	Single convolution	$\ell_{1.5}$	700	100 (FISTA)
2	Product-convolution	$\ell_{1.5}$	0.0135	55 (ADMM)
	Besson *et al.*	$\ell_{1.3}$	0.03	100 (FISTA)
	Single convolution	$\ell_{1.3}$	700	100 (FISTA)

TABLE III: Restoration parameters

D. In Vivo Results

For our *in vivo* experiments, we consider as benchmark method the one of Besson *et al.* [13] as their work is, to the best of our knowledge, the state-of-the-art on shift-variant US image restoration. To ensure the fairest comparison possible, we use their data, and reproduce their results using their code, which is available online.

For our method, we estimates 40 PSF arranged in a 4×10 grid and keep $K = 5$ kernels. Examples of projected PSF are illustrated in Figure 6. As can be seen the PSF is spatially-varying. The regularization strength parameter, λ, was tuned to maximize both SNR, TCR and visual quality. We set $\rho_1 = 20$ and $\rho_2 = 0.1$ for both carotids. We stop optimization when step-convergence is achieved, that is when $||\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_{k-1}||_2^2 / ||\hat{\mathbf{x}}_{k-1}||_2^2 \leq t$, where we set the tolerance $t = 10^{-6}$. Parameters common to our method and Besson method are presented in Table III.

Restoration results are shown in Figure 7. We see that product-convolution restoration produces images of overall visual quality similar to [13]. Yet, on Figure 7.c, we observe restoration artefacts along the south wall of the carotid (see red arrows), which are not present in Figure 7.b. Moreover, in the second carotid image (second row of Figure 7), the proposed method more effectively enhances fine details. To quantify results, we use the TCR and the Signal-to-Noise Ratio (SNR), which is not computed on the normalized envelope but on the linearized B-mode image, with:

$$\mathrm{SNR}(\mathbf{r}_b, \mathbf{r}_t) = \frac{|\mu_b - \mu_t|}{\sqrt{\sigma_b^2 + \sigma_t^2}}.$$

In addition to these two metrics used to ensure a fair comparison with Besson's results, which can be sensible as they are only computed on local patches (particularly the SNR metric as we can see from its strong variation across patches/images in Table IV), we incorporate the speckle-SNR, noted s-SNR [23]. It is more robust as it consider the entire image. The speckle-SNR is calculated on the envelope and is defined as the mean of the envelope divided by its standard deviation. Table IV reports the restoration metrics. The metrics are in line with the observations: the first carotid reconstructions (Figure 7.b-c) are of overall similar visual quality, whereas on the second carotid (Figure 7.f-g), product-convolution provides a better reconstruction, with less noise.

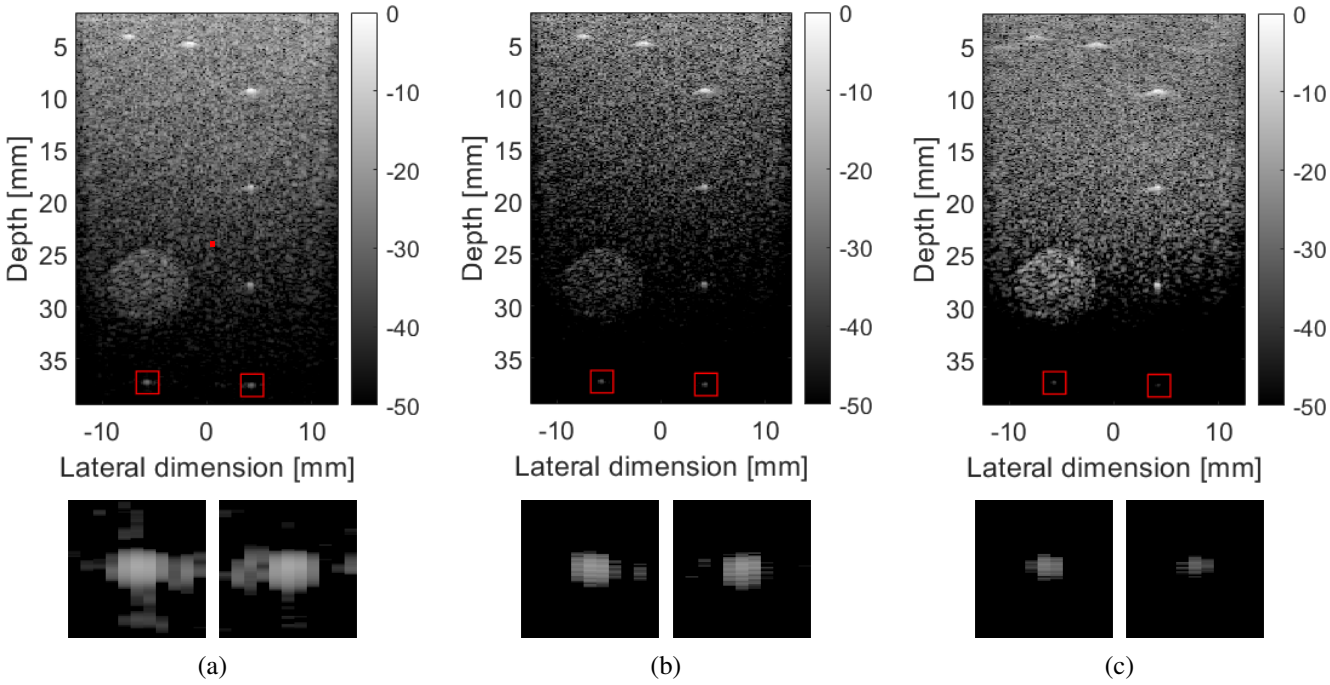

Fig. 5: (a): Phantom image; (b): Reconstruction with convolution; (c):Reconstruction with product-convolution.

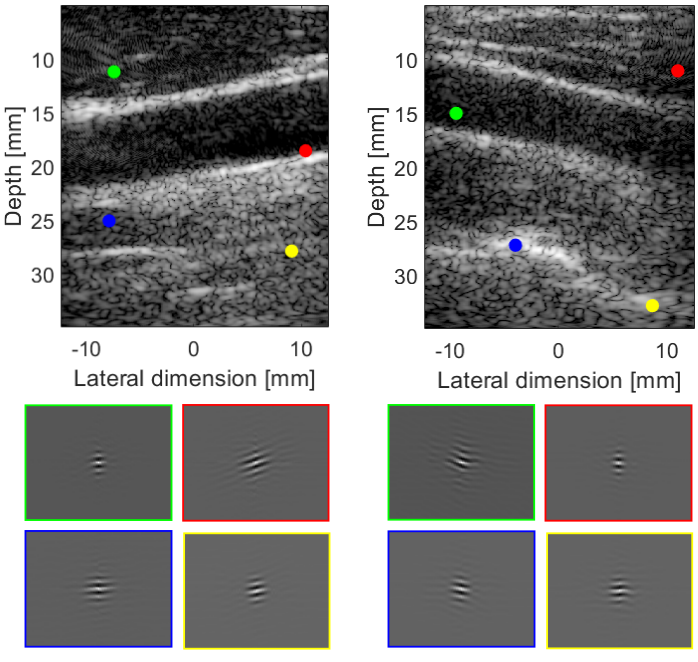

Fig. 6: Examples of projected PSF at different localisations

Carotid	Method	SNR green/red patches	TCR [dB] green/red p.	s-SNR
1	Prod.-conv.	18.99 / 33.71	**35.51 / 29.84**	**0.0154**
	Besson *et al.*	**19.80 / 53.4**	30.55 / 29.54	0.0140
	Single conv.	12.87 / 29.47	33.24 / 26.18	0.0150
2	Prod.-conv.	**62.01 / 39.60**	**22.95 / 30.54**	**0.0187**
	Besson *et al.*	28.25 / 23.41	19.59 / 30.50	0.0144
	Single conv.	26.11 / 23.05	17.20 / 24.40	0.0153

TABLE IV: In vivo results metrics

E. Computation Speed

In this section, we are interested in comparing the speed of the proposed restoration method against the approach of Besson *et al.* To quantify this, we measure the average restoration time of the first carotid over 10 runs on an Intel Core i5-10500 CPU @ 3.10GHz equipped with MatLab R2024a. We report the computation times in Table V. We see that our reconstructions are 19 to 30 times faster. Note that the variation in computation speed of the proposed method across samples is due to the different number of iterations required to reach convergence. A more robust measure is the time per iteration, for which we are 17 times faster.

Let us now explain this result. As explained in Section II-F, one ADMM iteration, using product-convolution, has a computational complexity of order $\mathcal{O}(2KN\log_2(N))$. On the other hand, one Fast Iterative Shrinkage Algorithm (FISTA) iteration using Besson *et al.* model has a complexity of order $\mathcal{O}(4N_eN)$, with N_e the number of probe elements. Although their model is asymptotically faster, with linear complexity, our log-linear model has a lower computational burden on real images such as the first carotid. It is of size 382×1228, giving $N = 496\,096$ and $\log_2(N) \approx 13.1$. We only keep 5 convolution kernels, resulting in a computational complexity of $\mathcal{O}(131N)$. In contrast, since the probe used to acquire the carotid image has 128 elements, the model of Besson *et al.* results in a computational complexity of order $\mathcal{O}(512N)$.

In general, as long as $K\log_2(N) < 2N_e$, our product-convolution model is computationally more efficient. To put this into perspective, with this real setting, i.e. $K = 5$ and $N_e = 128$, product-convolution remains faster for image up to $5.10^7 \times 5^7$. To the best of our knowledge, there is no real world settings where we encounter images of such size.

Although product-convolution allows for a lower computational burden than Besson *et al.* method in this setting, this alone does not explain such differences in computation time.

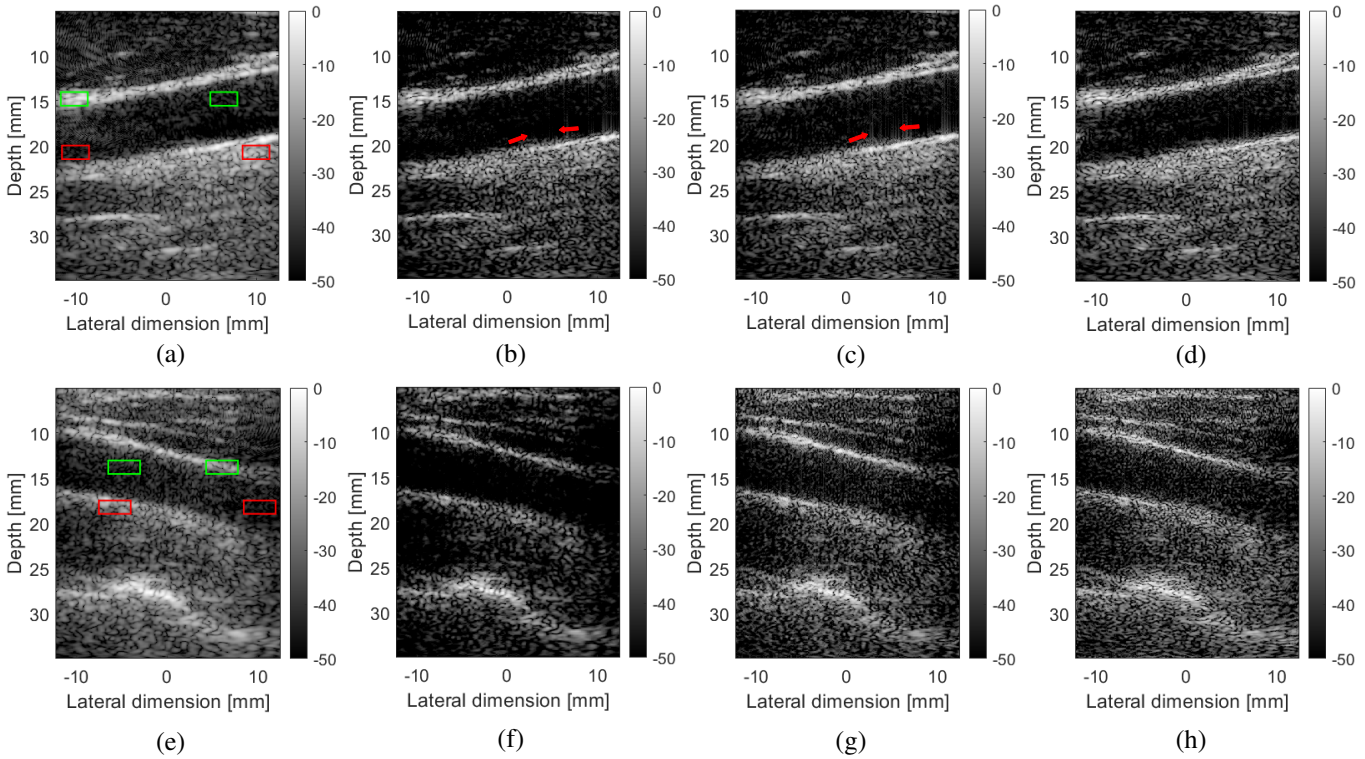

Fig. 7: (a): First carotid data with patches used for SNR and TCR; Restoration of the carotid using (b): product-convolution; (c): Besson *et al.* [13] model; (d): simple convolution model; (e) Second carotid data with patches; Restoration of the carotid using (f): product-convolution; (g): Besson *et al.* [13] model; (h): simple convolution model;

	Product-convolution ($K = 5$) Carotid 1 / Carotid 2	Besson *et al.*	Convolution
Total time (s)	42.4 / 26.5	798.8	8.8
Time per iteration (s)	0.45	7.99	0.08

TABLE V: Computation times of carotid restoration.

We also leverage the highly optimised built-in MatLab FFT, whereas Besson *et al.* code is unoptimized MatLab code.

IV. Discussion

We have proposed to use product-convolution, an approximation of shift-variant blur, as the direct model for US image reconstruction. We described a SVD-based strategy to construct the product-convolution operator based solely on input data. We then derived an efficient optimization scheme, using ADMM. We showed that our method is competitive with the state-of-the-art approaches.

Our work represent an alternative to existing method, by demonstrating that product-convolution is an efficient and effective direct model for US image restoration. In particular, our method has a lower computational burden on real data. It also relies on convolution, for which very efficient algorithms are freely available [24]. Beside, product-convolution can be used on US image beamformed with any beamformer, whereas the state-of-the-art method [13] assumes DAS beamforming.

Moving forward, there are two main leads to explore. The first one is related to the PSF estimation method. As explained in Section III-B2, if we were able to correctly estimate the PSF amplitude, product-convolution would be able to compensate attenuation, thus correcting Time Gain Compensation (TGC). At the moment, this is only possible on simulated data, using simulated PSF. The second lead concerns computation time. Taking inspiration from [25], real time US image restoration using product-convolution could be possible.

V. Conclusion

In this work, we tackle the US image restoration problem, taking into account the varying nature of the PSF. When the PSF is not assumed to be stationary, the image formation is modelled by a space-varying blur. We propose to approximate this blur using a fast approximation known as product-convolution. We present a SVD-based method to form the product-convolution operator based solely on input data. We also describe an ADMM splitting that results in efficient optimization. We finally show that our method obtain state-of-the-art results, while having lower computational time.

Appendix

A. Proof of Proposition 1

We start from the proximal operator definition:

$$\begin{aligned}\operatorname{prox}_{\frac{1}{\rho_1} f_1}(\mathbf{x}) &= \arg\min_{\mathbf{z}} \frac{1}{2}\|\mathbf{H}\mathbf{z}-\mathbf{y}\|_2^2 + \frac{\rho_1}{2}\|\mathbf{z}-\mathbf{x}\|_2^2 \\ &= (\mathbf{H}^T\mathbf{H} + \rho_1 \mathbf{I}_{KN})^{-1}(\mathbf{H}^T\mathbf{y} + \rho_1 \mathbf{x}).\end{aligned} \tag{20}$$

Using the Woodbury identity:

$$(\mathbf{A}+\mathbf{U}\mathbf{C}\mathbf{V})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1}+\mathbf{V}\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{A}^{-1},$$

we have (by setting $\mathbf{A}=\rho_1\mathbf{I}_{KN}$, $\mathbf{U}=\mathbf{H}^T$, $\mathbf{C}=\mathbf{I}_N$ and $\mathbf{V}=\mathbf{H}$):

$$(\mathbf{H}^T\mathbf{H}+\rho_1\mathbf{I}_{KN})^{-1}=\frac{1}{\rho_1}\mathbf{I}_{KN}-\frac{1}{\rho_1^2}\mathbf{H}^T(\mathbf{I}_N+\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{H}. \quad (21)$$

One can observe that $\mathbf{H}\mathbf{H}^T$ is diagonal in the Fourier domain:

$$\mathbf{H}\mathbf{H}^T=\mathbf{F}_N^T\mathbf{\Lambda}\mathbf{F}\mathbf{F}^T\mathbf{\Lambda}^T\mathbf{F}_N=\mathbf{F}_N^T\sum_{k=1}^{K}\mathbf{\Lambda}_k\mathbf{\Lambda}_k^T\mathbf{F}_N. \quad (22)$$

To simplify notations, let $\mathbf{D}=\mathbf{I}_N+\sum_{k=1}^{K}\mathbf{\Lambda}_k\mathbf{\Lambda}_k^T$. Injecting (22) into (21) gives:

$$\begin{aligned}(\mathbf{H}^T\mathbf{H}+\rho_1\mathbf{I}_{KN})^{-1}&=\frac{1}{\rho_1}\mathbf{I}_{KN}-\frac{1}{\rho_1^2}\mathbf{H}^T\mathbf{F}_N^T\mathbf{D}^{-1}\mathbf{F}_N\mathbf{H}\\&=\frac{1}{\rho_1}\mathbf{I}_{KN}-\frac{1}{\rho_1^2}\mathbf{F}^T\mathbf{\Lambda}^T\mathbf{F}_N\mathbf{F}_N^T\mathbf{D}^{-1}\mathbf{F}_N\mathbf{F}_N^T\mathbf{\Lambda}\mathbf{F}\\&=\frac{1}{\rho_1}\mathbf{I}_{KN}-\frac{1}{\rho_1^2}\mathbf{F}^T\mathbf{\Lambda}^T\mathbf{D}^{-1}\mathbf{\Lambda}\mathbf{F},\end{aligned}$$

which completes the proof.

References

[1] V. Perrot *et al.*, "So you think you can das? a viewpoint on delay-and-sum beamforming," *Ultrasonics*, vol. 111, p. 106 309, 2021.

[2] J. Ng *et al.*, "Modeling ultrasound imaging as a linear, shift-variant system," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control.*, vol. 53, no. 3, pp. 549–563, 2006.

[3] J. A. Jensen, "Deconvolution of ultrasound images," *Ultrasonic imaging*, vol. 14, no. 1, pp. 1–15, 1992.

[4] C. Yu *et al.*, "A blind deconvolution approach to ultrasound imaging," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control.*, vol. 59, no. 2, pp. 271–280, 2012.

[5] X. Li *et al.*, "Deconvolution based on sparsity and continuity improves the quality of ultrasound image," *Comput. Biol. Med.*, vol. 169, p. 107 860, 2024.

[6] F. Duelmer *et al.*, "Phocus: Physics-based deconvolution for ultrasound resolution enhancement," in *International Workshop on Advances in Simplifying Medical Ultrasound*, Springer, 2024, pp. 35–44.

[7] L. Denis *et al.*, "Fast approximations of shift-variant blur," *Int. J. Comput. Vis.*, vol. 115, pp. 253–278, 2015.

[8] P. Escande and P. Weiss, "Approximation of integral operators using product-convolution expansions," *J. Math. Imaging Vis.*, vol. 58, pp. 333–348, 2017.

[9] O. V. Michailovich *et al.*, "A novel approach to the 2-d blind deconvolution problem in medical ultrasound," *IEEE Trans. Med. Imaging.*, vol. 24, no. 1, 2005.

[10] M. I. Florea *et al.*, "An axially variant kernel imaging model applied to ultrasound image reconstruction," *IEEE Signal Process. Lett.*, vol. 25, no. 7, 2018.

[11] M. I. Florea *et al.*, "Restoration of ultrasound images using spatially-variant kernel deconvolution," in *2018 IEEE ICASSP*, IEEE, 2018, pp. 796–800.

[12] J. A. Jensen, "Simulation of advanced ultrasound systems using field ii," in *IEEE ISBI: Nano to Macro*, IEEE, 2004, pp. 636–639.

[13] A. Besson *et al.*, "A physical model of nonstationary blur in ultrasound imaging," *IEEE Trans. Comput. Imaging.*, vol. 5, no. 3, pp. 381–394, 2019.

[14] V. Debarnot *et al.*, "Learning low-dimensional models of microscopes," *IEEE Trans. Comput. Imaging.*, vol. 7, pp. 178–190, 2020.

[15] R. C. Flicker and F. J. Rigaut, "Anisoplanatic deconvolution of adaptive optics images," *JOSA A*, vol. 22, no. 3, pp. 504–513, 2005.

[16] E. O. Brigham and R. E. Morrow, "The fast fourier transform," *IEEE spectrum*, vol. 4, no. 12, pp. 63–70, 1967.

[17] R. Sibson, "A brief description of natural neighbour interpolation," *Interpreting Multivariate Data*, pp. 21–36, 1981.

[18] D. Garcia, "Simus: An open-source simulator for medical ultrasound imaging. part i: Theory & examples," *Comput. Methods Programs Biomed.*, vol. 218, p. 106 726, 2022.

[19] M. Alessandrini *et al.*, "A restoration framework for ultrasonic tissue characterization," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control.*, vol. 58, no. 11, 2011.

[20] S. Boyd *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[21] E. Soubies *et al.*, "Pocket guide to solve inverse problems with globalbioim," *Inverse Problems*, vol. 35, no. 10, p. 104 006, 2019.

[22] R. A. Schoop *et al.*, "Experimental validation of ultrasound beamforming with end-to-end deep learning for single plane wave imaging," *arXiv:2404.14188*, 2024.

[23] O. V. Michailovich *et al.*, "Despeckling of medical ultrasound images," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control.*, vol. 53, no. 1, pp. 64–78, 2006.

[24] M. Frigo and S. G. Johnson, "The design and implementation of FFTW3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005, Special issue on "Program Generation, Optimization, and Platform Adaptation".

[25] P. Escande and P. Weiss, "Accelerating l_1-l_2 deblurring using wavelet expansions of operators," *J. Comput. Appl. Math.*, vol. 343, pp. 373–396, 2018.