Research Paper

Algorithm for Computationally Efficient Imaging of Sound Speed in Conventional Ultrasound Sonography

Piotr KARWAT

Department of Ultrasound, Institute of Fundamental Technological Research
Polish Academy of Sciences
Warsaw, Poland; e-mail: pkarwat@ippt.pan.pl

The speed of sound (SoS) in tissues reflects their mechanical properties and therefore can carry valuable diagnostic information. In conventional ultrasound sonography (US), however, this information is not readily available. Furthermore, since the actual SoS is unknown, image reconstruction is carried out using an average SoS value for soft tissues. The resulting local deviations from the actual SoS lead to aberrations in US images. Methods for SoS imaging in US therefore have the potential to enable the correction of aberrations in classical US. In addition, they could also become a new US modality.

There are several approaches to SoS image reconstruction. They differ in terms of input data requirements, computational complexity, imaging quality, and the potential for signal analysis at the intermediate stages of processing. This article presents an algorithm with multi-stage processing and low computational complexity.

The algorithm was verified through numerical simulations and phantom measurements. The obtained results show that it can correctly estimate SoS in layered media, which in most cases model the tissue structure well. With its computational complexity of $O(n)$, the algorithm can be implemented in real-time ultrasound imaging systems with limited hardware performance, such as portable ultrasound devices.

Keywords: speed of sound; ultrasound imaging; computational complexity.

1. Introduction

The speed of sound (SoS) in tissues reflects their mechanical properties and depends on many aspects. Various soft tissue types can be characterized by an SoS ranging approximately from 1440 m/s in fat to 1620 m/s in skin (COBBOLD, 2007; Foundation for Research on Information Technologies in Society, n.d.). Moreover, the SoS can vary with tissue composition. For example, excess fat content leads to a decreased SoS, as in fatty liver (GHOSHAL *et al.*, 2012). It can also be altered due to pathological changes in the extracellular matrix, specifically the excessive accumulation of collagen and the development of fibrosis, which often accompany inflammatory diseases and tumors.

The SoS can therefore provide information on tissue type and condition. It has already proven usable in the assessment of breast tumor malignancy in ultrasound computed tomography (UCT) (ANDRE *et al.*, 2012). This technique, however, requires that the examined tissue is accessible from all sides in the imaging plane, and preferably contains no bones. As a result, UCT has a very narrow field of application, limited mainly to breast imaging. Implementing the SoS modality into conventional ultrasound sonography (US) could allow for utilizing the SoS information on a much larger scale and in the context of many other organs and diseases.

The existing solutions for SoS imaging in conventional US are relatively new. JAEGER *et al.* (2014; 2015) introduced a mathematical model and an algorithm for SoS reconstruction, which they called the computed ultrasound tomography in echo mode (CUTE). It utilized phase differences between complex images obtained for a number of plane wave transmissions at different angles. The authors proposed a model describing these phase differences as a function of SoS local errors made in image reconstruction. To solve the inverse problem, they proposed the use of the pseudoinverse of the transformation matrix based on their model. At first, the processing was conducted in the frequency domain, which made the algorithm compu-

tationally efficient. However, it also made it impossible to consider imaging dead zones, which in turn led to bias in the resulting SoS estimates. To address this problem, a spatial domain version of CUTE processing was proposed (JAEGER, FRENZ, 2015; SANABRIA _et al._, 2018). This led to an improvement in the quality of resulting SoS images to the point where CUTE could be referred to as a quantitative method. This improvement, however, was accompanied by a significant increase in computational complexity. In another paper (STÄHLI _et al._, 2020), the authors indicated that certain assumptions made in the CUTE method are wrong. As a solution, they proposed a new algorithm and a modification in the input data to ensure the same transmit-receive mid-angle in the compared images. In exchange for more complex preprocessing and a reduced resolution, this method offers more accurate SoS estimates.

Another group of methods for SoS imaging uses convolutional neural networks (CNNs). In some studies (FEIGIN _et al._, 2020; YOUNG _et al._, 2022), CNNs were trained using raw echo data obtained through numerous simulations with the k-Wave toolbox (TREEBY, COX, 2010) for Matlab. SoS imaging using CNNs does not require the initial image reconstruction. Moreover, the reported models demonstrated good performance with data from just three transmissions, a notably smaller number to what is reported in the case of CUTE methods. Finally, CNNs are fast enough to be implemented in real-time imaging systems. On the other hand, analyzing and understanding the CNN's way of interpreting the signal in order to find a correlation with some physical model poses challenges for researchers.

In (KARWAT, 2019), the author proposed a modification to the CUTE mathematical model and introduced a corresponding SoS reconstruction algorithm based on it. This modification made it possible to solve the inverse problem at low computational cost and was therefore called Quick-CUTE (Q-CUTE). The algorithm was validated using simulation data.

In this study, the input signal of the CUTE method is analyzed with respect to its compliance with the modified mathematical model. Moreover, the algorithm is described in more detail, including an analysis of the signal at individual stages of the processing. Finally, the results of the measurement-based validation are presented.

2. Methodology

2.1. General CUTE model

The general idea behind the spatial domain CUTE method is presented in (JAEGER, FRENZ, 2015; SANABRIA _et al._, 2018), but for clarity, it will also be explained here. Reconstruction of images in conventional

US is usually performed according to the delay-and-sum (DAS) algorithm that includes, inter alia, compensation for the ultrasound pulse propagation delays t. The calculation of these delays requires knowledge of the SoS value c along the pulse propagation path r from the probe to the pixel of interest and back to the probe:

$$t = \int_r \frac{\mathrm{d}r}{c}. \tag{1}$$

However, the value of c is not known precisely, therefore its approximation is used. Typically, an average SoS value for soft tissues is used for this purpose. Any discrepancies between the SoS value c_R adopted for reconstruction and the actual SoS spatial distribution c result in calculated propagation delays t_r being different from the actual delays t. Consequently, these discrepancies lead to time delay errors τ:

$$\tau = t - t_r = \int_r \frac{\mathrm{d}r}{c} - \int_r \frac{\mathrm{d}r}{c_R}$$
$$= \int_r \sigma \, \mathrm{d}r - \int_r \sigma_R \, \mathrm{d}r = \int_r \Delta\sigma \, \mathrm{d}r. \tag{2}$$

For notation simplicity, σ, σ_R, and $\Delta\sigma$ terms are introduced in the aforementioned equation. The first two represent actual and approximated values of sound slowness (inverse of SoS), respectively. The last one stands for the error in sound slowness:

$$\Delta\sigma(z,x) = \sigma(z,x) - \sigma_R = \frac{1}{c(z,x)} - \frac{1}{c_R}. \tag{3}$$

Coming back to the time delay errors τ, they lead to phase aberrations φ in the reconstructed complex radio-frequency (CRF) images. These phase aberrations cannot be measured directly for an individual CRF image but can be estimated as local phase differences $\Delta\varphi$ with respect to another CRF image acquired for different propagation paths r. Based on the $\Delta\varphi$, however, only relative values of τ can be obtained. Therefore, instead of τ, time delay error differences $\Delta\tau$ will be further considered:

$$\Delta\tau_{m,n} = \tau_n - \tau_m = \int_{r_n} \Delta\sigma \, \mathrm{d}r - \int_{r_m} \Delta\sigma \, \mathrm{d}r, \tag{4}$$

where m and n subscripts identify the m-th and n-th images being compared. The above equation describes the forward problem, i.e., it defines the observed $\Delta\tau$ as a function of the error in sound slowness $\Delta\sigma$. The CUTE algorithms are designed to solve the inverse problem, that is to estimate the SoS corrections based on the observed $\Delta\tau$. To make this feasible, some assumptions are made. First, similarly to the classical DAS reconstruction algorithm, the concept of rays known from geometrical optics is used to describe the propagation of ultrasound. Given that relative SoS variations in soft tissues are relatively low (up to approximately $\pm6\%$ from the SoS mean value), it is also assumed that the ultrasound rays do not refract.

Furthermore, if the receive aperture and apodization used for reconstructing each pixel remain the same for the images being compared, the parts of time delay errors τ_m and τ_n associated with the return paths (from pixel to probe) are assumed to be equal. The time delay error differences $\Delta\tau_{m,n}$ are therefore a consequence of the errors committed on the transmit paths (from probe to pixel) only, as the return paths errors cancel each other out. With the above assumptions, and bearing in mind that CUTE methods typically use images obtained for plane wave emissions at various angles θ, r represents a rectilinear transmit propagation path taken at angle θ from a probe to a pixel of interest P.

The formulation of the forward problem in Eq. (4) can be further modified. Its frequency domain version is utilized in the first reported version of CUTE (JAEGER *et al.*, 2014; 2015). Its original form is the foundation for the spatial domain CUTE algorithm (JAEGER, FRENZ, 2015; SANABRIA *et al.*, 2018). Finally, it can be linearly approximated to obtain a common integration path, as in Q-CUTE (KARWAT, 2019). Depending on the formulation of the forward problem, solving the inverse problem can be conducted in various ways. This, in turn, leads to multiple CUTE versions differing in terms of SoS imaging quality and computational complexity.

2.2. Spatial domain CUTE processing

The model described in Eq. (4) is the basis for the spatial domain CUTE algorithm. It is used for the determination of the forward problem (calculation of $\Delta\tau$ based on $\Delta\sigma$) transformation matrix. Next, its pseudo-inverse is computed for subsequent use in solving the inverse problem (calculation of $\Delta\sigma$ based on $\Delta\tau$). Due to the ill-conditioning of the inverse problem, spatial gradient regularization is used.

Having the pre-calculated inverse transformation matrix, it only takes $\Delta\tau$ to compute $\Delta\sigma$. The time delay errors $\Delta\tau_{m,n}$ are estimated from the local phase differences $\Delta\varphi_{m,n}$ between CRF_m and CRF_n complex images as well as the signal frequency f:

$$\Delta\tau_{m,n} = \frac{\Delta\varphi_{m,n}}{2\pi f}. \tag{5}$$

Due to the interference nature of the CRF images, to obtain a usable signal, the local phase differences $\Delta\varphi_{m,n}$ are estimated according to the equation:

$$\Delta\varphi_{m,n} = \arg\left[\mathrm{filt}\left(CRF_n \circ CRF_m^*\right)\right], \tag{6}$$

where * and $\circ$ operators are the complex conjugate and Hadamard (element-wise) product, respectively. The term "filt" denotes spatial smoothing filtration, and "arg" returns the arguments of complex numbers.

The $\Delta\tau$ maps obtained using the above methodology are of poor lateral resolution due to the absence of transmit focusing in the CRF images. In order to enhance the resolution, the input CRF images can be obtained through coherent compounding for a number of closely spaced transmit angles θ. Another issue is the aliasing and decoherence, which may occur if the difference between θ_m and θ_n angles is substantial. On the other hand, if the difference between θ_m and θ_n is small, the amplitude of the useful part of $\Delta\tau$ is low compared to the interference part, adversely affecting the sensitivity of the method. To avoid the aliasing and limit the decoherence while maintaining the sensitivity, $\Delta\tau_{m,n}$ is estimated using a number of closely spaced θ angles that cover the angular space between the target pair of θ_m and θ_n angles.

To illustrate it, let us consider the objective of obtaining $\Delta\tau_{m,n}$ map for transmit angles $[\theta_m, \theta_n] = [0°, 8°]$. To avoid the aliasing and preserve the coherence of the paired CRF images, one would reduce the difference in θ from 8° to 2°, and this involves using a set of CRF images for transmit angles $\theta = \{0°, 2°, 4°, 6°, 8°\}$. For each pair of consecutive θ angles, $\Delta\tau$ is calculated. The final $\Delta\tau_{m,n}$ is the sum of the $\Delta\tau$ maps obtained for θ pairs $\{[0°, 2°], [2°, 4°], [4°, 6°], [6°, 8°]\}$. Furthermore, in order to enhance the $\Delta\tau$ lateral resolution, each CRF image would be obtained through coherent compounding for angles, e.g., being $\{-1°, -0.5°, 0°, 0.5°, 1°\}$ around the base angle θ. This means that a single $\Delta\tau_{m,n}$ map would require plane wave transmissions at angles in the range $[-1°, 9°]$ with a step of 0.5°, which amounts to 21 transmissions.

More details on the spatial domain CUTE algorithm can be found in the presentation by JAEGER *et al.* (2015) and the paper by SANABRIA *et al.* (2018). For simplicity, in the remaining sections of this paper, the term CUTE will be used to refer to the spatial domain version of the CUTE method.

2.3. Q-CUTE model

The method being the subject of this article is the Q-CUTE technique, presented briefly in (KARWAT, 2019). It is based on a model that is a modified version of the general CUTE model described by Eq. (4). The aim of the modification is to obtain a common integration path. This, in turn, enables a different approach to the input signal $\Delta\tau$ and opens new ways of solving the inverse problem. Let us start from the beginning, though.

Because $\mathrm{d}r = \mathrm{d}z'/\cos\theta$, Eq. (4) can be rewritten as:

$$\Delta\tau_{m,n}(z,x) = \frac{1}{\cos\theta_n}\int_0^z \Delta\sigma\left(r_n\right)\mathrm{d}z'$$

$$-\frac{1}{\cos\theta_m}\int_0^z \Delta\sigma\left(r_m\right)\mathrm{d}z'. \tag{7}$$

Let us approximate linearly $\Delta\sigma$ along r_m and r_n paths using $\Delta\sigma$ and its x-derivative $\Delta\sigma_x$ along a path $r_{m,n}$ that is horizontally equidistant from r_m and r_n, as shown in Fig. 1.
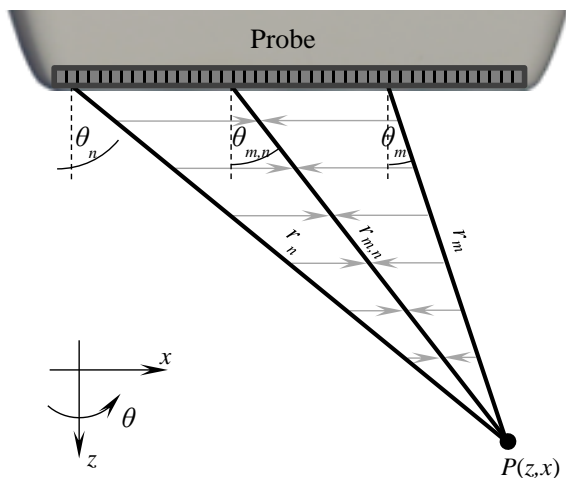

Fig. 1. Sketch of the linear approximation approach used in the Q-CUTE model. The approximation is done horizontally as indicated by gray arrows.

The corresponding angle $\theta_{m,n}$ equals:

$$\theta_{m,n} = \tan^{-1}\left(\frac{\tan\theta_m + \tan\theta_n}{2}\right), \qquad (8)$$

and the linear approximations of $\Delta\sigma$ along the r_m and r_n paths are:

$$\Delta\sigma(r_m) \approx \Delta\sigma(r_{m,n})$$
$$+ \Delta\sigma_x(r_{m,n})(z-z')\left(\frac{\tan\theta_n - \tan\theta_m}{2}\right),$$
$$\Delta\sigma(r_n) \approx \Delta\sigma(r_{m,n})$$
$$- \Delta\sigma_x(r_{m,n})(z-z')\left(\frac{\tan\theta_n - \tan\theta_m}{2}\right). \qquad (9)$$

With this modification, the $r_{m,n}$ becomes the new, common integration path, and Eq. (7) takes the form:

$$\Delta\tau_{m,n}(z,x) = a_{m,n} \int_0^z \Delta\sigma\,(r_{m,n})\,\mathrm{d}z'$$
$$+ b_{m,n} \int_0^z (z'-z)\Delta\sigma_x\,(r_{m,n})\,\mathrm{d}z'. \qquad (10)$$

Terms $a_{m,n}$ and $b_{m,n}$ in Eq. (10) are functions of θ_m and θ_n:

$$a_{m,n} = \left(\frac{1}{\cos\theta_n} - \frac{1}{\cos\theta_m}\right),$$
$$b_{m,n} = \left(\frac{1}{\cos\theta_n} + \frac{1}{\cos\theta_m}\right)\left(\frac{\tan\theta_n - \tan\theta_m}{2}\right), \qquad (11)$$
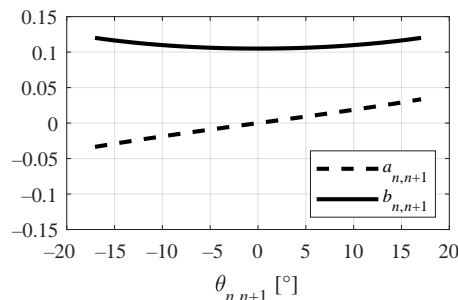
and are shown in Fig. 2.

Fig. 2. Graphs of $a_{m,n}$ and $b_{m,n}$ terms as functions of $\theta_{m,n}$.

Integration by parts of the second integral in Eq. (10), after some rearrangements, yields:

$$\Delta\tau_{m,n}(z,x) = a_{m,n} \int_0^z \Delta\sigma(r_{m,n})\,\mathrm{d}z'$$
$$- b_{m,n} \iint_0^z \Delta\sigma_x(r_{m,n})\,\mathrm{d}z'^2. \qquad (12)$$

The aforementioned forward problem model is the basis for the Q-CUTE method.

2.4. Q-CUTE processing

The estimation of $\Delta\tau$ in the Q-CUTE method is conducted in the same way as in CUTE, i.e., according to Eqs. (5) and (6). Solving the inverse problem, in turn, is done in two steps.

First, by using the properties of $a_{m,n}$ and $b_{m,n}$, i.e., parity and amplitude ratio, especially for $\theta_{m,n}$ close to zero (Fig. 2), Eq. (12) is simplified by neglecting the first integral on the right side of the equation:

$$\Delta\tau_{m,n}(z,x) \approx - b_{m,n} \iint_0^z \Delta\sigma_x(r_{m,n})\,\mathrm{d}z'^2. \qquad (13)$$

This allows a direct calculation of $\Delta\sigma_x$ for each $[m,n]$ pair:

$$\Delta\sigma_x\,(z,x,[m,n]) \approx \frac{-1}{b_{m,n}}\left[\frac{d^2}{\mathrm{d}z'^2}\Delta\tau_{m,n}(r_{m,n})\right]_{z'=z}. \qquad (14)$$

The $\Delta\sigma_x$ is next averaged over $[m,n]$ pairs, yielding $\overline{\Delta\sigma_x}$. This, in turn, is integrated with respect to x to obtain the estimate of $\Delta\sigma$:

$$\Delta\sigma(z,x) = \int_{-\infty}^x \overline{\Delta\sigma_x}\,(z,x')\,\mathrm{d}x' + C(z), \qquad (15)$$

where C is the constant of integration (it is constant along the x-direction, but may vary with z).

In the second step of the Q-CUTE algorithm, the missing $C(z)$ is determined. The formula for C can be derived using the unmodified form of Eq. (12). Substituting $\overline{\Delta\sigma_x}$ for $\Delta\sigma_x$, and expanding $\Delta\sigma$ according to

Eq. (15), after rearrangements, yields the C estimates based on individual $[m,n]$ pairs:

$$C_{m,n}(z,x) = \frac{1}{a_{m,n}}\left[\frac{d}{dz'}\Delta\tau_{m,n}(r_{m,n})\right]_{z'=z}$$
$$- \int_{-\infty}^{x}\overline{\Delta\sigma_x}(z,x')\,dx'$$
$$+ \frac{b_{m,n}}{a_{m,n}}\int_{0}^{z}\overline{\Delta\sigma_x}(r_{m,n})\,dz'. \qquad (16)$$

To obtain the final estimate for C, the values calculated using the above formula need to be averaged over $[m,n]$ pairs and the x-dimension. However, for $\theta_{m,n}$ close to zero, $a_{m,n}$ is close to zero as well. This means that $\Delta\tau_{m,n}$ for small $|\theta_{m,n}|$ does not provide much information on C, and dividing by $a_{m,n}$, in this case, would only amplify distortions in $\frac{d}{dz}\Delta\tau_{m,n}$. To prevent this, the averaging over $[m,n]$ pairs uses $a_{m,n}^2$ as weights, which leads to inverse variance weighting.

Finally, knowing the $\Delta\sigma$ (including the constant of integration C), one can calculate the speed of sound c using a reorganized version of Eq. (3):

$$c(z,x) = \frac{c_0}{1 + c_0\Delta\sigma(z,x)}. \qquad (17)$$

2.5. Regularized derivative operator

As in the case of the CUTE algorithm, Q-CUTE needs regularization to prevent amplification of interference present in the $\Delta\tau$ data. In the Q-CUTE algorithm, this is realized by replacing the derivative operators that act on $\Delta\tau$ with regularized ones. As a matrix model $\mathbf{D}_{\text{reg}}$ of the regularized derivative operator, a pseudo-inverse of integration operator matrix $\mathbf{J}$ (Fig. 3a) with gradient regularization (gradient operator $\mathbf{D}$ shown in Fig. 3b) was adopted:

$$\mathbf{D}_{\text{reg}} = \left(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{D}^T\mathbf{D}\right)^{-1}\mathbf{J}^T, \qquad (18)$$

where λ denotes the regularization factor. Figure 3c shows an exemplary $\mathbf{D}_{\text{reg}}$ matrix, while Fig. 3d presents selected impulse responses from $\mathbf{D}_{\text{reg}}$.
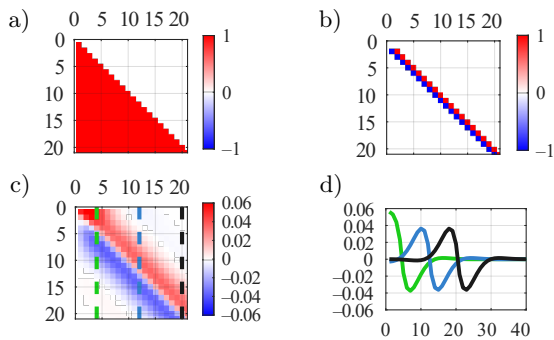

Fig. 3. Examples of a) integration operator matrix $\mathbf{J}$; b) gradient operator matrix $\mathbf{D}$; c) regularized derivative operator matrix $\mathbf{D}_{\text{reg}}$; d) selected impulse responses of $\mathbf{D}_{\text{reg}}$. The first quarter of matrices is shown in a)–c).

These types of impulse responses can be precisely approximated using pairs of computationally efficient infinite impulse response (IIR) filters (Lyons, 2004) operating one in forward and the other backward direction. Implementation of the regularized derivative operator depicted in Figs. 3c and 3d requires the use of second-order IIR filters. The IIR filter coefficients were precomputed to approximate the central, antisymmetric impulse response of $\mathbf{D}_{\text{reg}}$ for a chosen value of λ. However, as shown in Figs. 3c and 3d, impulse responses close to signal boundaries are more complex. To reflect the changes in the $\mathbf{D}_{\text{reg}}$ impulse responses and thus ensure proper derivative operation when approaching signal boundaries, the output of each filter is multiplied by a precalculated correction vector. The vector compensates for two effects resulting from clipping the forward or backward impulse responses. Firstly, it equalizes the areas under both impulse responses so that the derivative of a constant signal is always 0. Secondly, it corrects the change in the distance between the centers of mass of the impulse responses so that the derivative of a linear signal is always constant.

2.6. Computational complexity

The implementation of preprocessing, i.e., the estimation of $\Delta\tau$ based on a set of CRF images, can be done in a variety of ways. They can differ in terms of optional CRF images compounding, pairing strategy, or type and order of the smoothing filter used in Eq. (6). As a result, the discussion of the computational complexity of the preprocessing becomes hindered. However, the preprocessing for the Q-CUTE algorithm is the same as in other CUTE methods based on the model presented in Eq. (4). Hence, it does not introduce computational differences between the compared SoS imaging algorithms. Therefore, the description of computational complexity is limited to the subsequent processing.

Let us define n_{PIX} as the number of pixels in a single $\Delta\tau$ map, and $n_{\Delta\tau}$ as the number of $\Delta\tau$ maps. Then the size of the $\Delta\tau$ to $\Delta\sigma$ transformation matrix in the CUTE algorithm is $n_{\text{PIX}} \times (n_{\text{PIX}} \cdot n_{\Delta\tau})$. Although the forward problem transformation matrix is sparse, the inverse problem transformation matrix is not. The CUTE computational complexity, expressed as a number of multiply-add operations, is therefore equal $n_{\Delta\tau} \cdot n_{\text{PIX}}^2$.

In the case of Q-CUTE algorithms, the computational complexity is a sum of the complexities of their components. Apart from the operations directly represented in the equations (derivatives, integrations, and weighted sums), they also include 1-D linear interpolation every time some operation is performed along a direction that is not aligned with rows or columns of the data array. The computational cost of each operation

and their total cost in the Q-CUTE algorithms are presented in Table 1. The total computational complexity of the Q-CUTE algorithm is equal to $35 \cdot n_{\Delta\tau} \cdot n_{\text{PIX}}$.

Table 1. List of the operations, their computational cost per pixel of the $\Delta\tau$ map, and the number of calls per single $\Delta\tau$ map in the Q-CUTE algorithm.

Operation name	Unit cost	# of calls	Cost
1-D linear interpolation	2	3	6
Derivative (regularized) (two 2nd order IIR filters)	12	2	24
Integration	1	1	1
Weighted sum	1	4	4
Total			**35**

2.7. Validation/input data

The Q-CUTE algorithm was tested and validated based on data obtained from phantom measurements as well as from numerical simulations.

Measurements were carried out with the use of the us4R-lite system (us4us, Warsaw, Poland) (CACKO, LEWANDOWSKI, 2022) and a linear array probe SL1543 (Esaote, Genoa, Italy) with 192 elements, a pitch of 0.245 mm, and a center frequency of 8 MHz. The data were acquired from a phantom model 1438 (Dansk Fantom Service, Frederikssund, Denmark). The main material of this phantom is characterized by an SoS of 1540 m/s. The phantom also contains four cylindrical targets, 10 mm in diameter, with SoS values equal to 1480, 1510, 1570, and 1600 m/s.

Numerical simulations were performed with probe parameters identical to those for SL1543, using a custom-developed MATLAB-based simulator tool. The simulator follows a similar principle to the Field software (JENSEN, 1996), i.e., it is based on linear acoustics and uses the concept of scattering points. Unlike Field, however, it can simulate any SoS maps (refraction is still omitted). Furthermore, it makes possible to simulate a medium in which the SoS map is different for the transmit and receive paths, which also distinguishes it from the k-Wave toolbox (TREEBY, COX, 2010). This feature proves useful in the separation of $\Delta\tau$ signal components presented in the Results section.

In both cases, i.e., measurements and simulations, the raw echo data were acquired using the single-element synthetic transmit aperture (SSTA) technique, elsewhere referred to as full matrix capture (FMC). It involves the separate transmission through each individual probe element and receiving the echoes each time with the entire probe. This strategy was used in the case of changing the transmission angles θ or SoS value c_R, because it does not require repeating measurements or simulations, just the FMC data need to be properly converted into a form corresponding to the plane wave imaging.

The obtained raw echo data underwent processing through a digital down converter (DDC), i.e., they were quadrature demodulated, low-pass filtered and decimated. The resulting complex echo signal was next reconstructed into CRF images using the delay-and-sum (DAS) algorithm with phase-error-free quadrature sampling (CHANG *et al.*, 1993).

The CRF images were reconstructed for diverging waves, so they had to be converted into a form corresponding to plane waves. This was done through the proper recombination of the CRF images. This conversion was done for transmit angles θ ranging from $-17°$ to $+17°$ in a step of 0.25°, resulting in a set of 137 CRF images. In the next step, a subset of CRF images with θ in a $\pm 2°$ range around a selected θ was coherently summed. The selected θ were in the range from $-15°$ to $+15°$ in a step of 2°, resulting in a set of 16 compounded CRF images. Then, based on pairs of compounded CRF images with consecutive θ values, $\Delta\tau$ maps were estimated using Eqs. (5) and (6). Next, the $\Delta\tau$ maps were summed in groups so that $\theta_n - \theta_m$ equaled 6°, which resulted in a set of five $\Delta\tau$ maps further processed to obtain a single SoS output image. A diagram of this procedure for a single $\Delta\tau$ map is presented in Fig. 4.
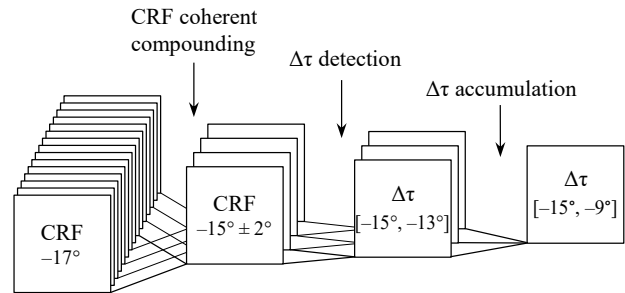

Fig. 4. Diagram of processing a set of CRF images to obtain a single $\Delta\tau$ map.

The SoS spatial distributions used in this study included uniform cases, horizontal layer cases, and circular inclusion cases. The layers and circular inclusions in measurement data were obtained through positioning the probe along and across cylindrical objects in the phantom, respectively.

In order to assess the SoS imaging quality, the results of the Q-CUTE method were compared with those obtained from the spatial domain CUTE algorithm considered as a reference at this stage of the research.

The appearance of the resulting SoS images depends on the regularization factors (CUTE) and the corresponding filters in the regularized derivative operators (Q-CUTE). Increasing the regularization factor results in smoother SoS images. The level of regularization was therefore a compromise between reducing the variance and maintaining the details of the SoS

images. It was also adjusted so that the average image of a simulated circular inclusion was reproduced with similar amplitudes and slopes by both algorithms.

All the raw echo data simulations and conversions, CRF image reconstructions, CUTE processing, and preparation of the results were executed using MAT-LAB 2021b (Mathworks, Inc., Natick, Massachusetts).

3. Results

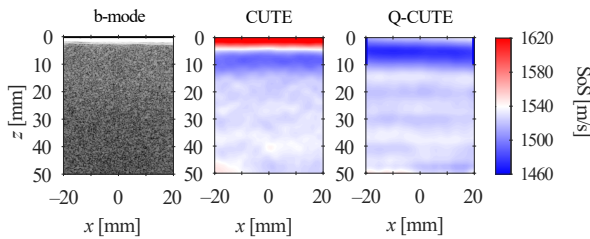The SoS images reconstructed using the CUTE and Q-CUTE algorithms are depicted in Figs. 5–8. Figure 5

Fig. 5. Images of the homogeneous 1540 m/s part of the phantom: b-mode (left), SoS images obtained with the use of CUTE (middle), and with Q-CUTE (right).
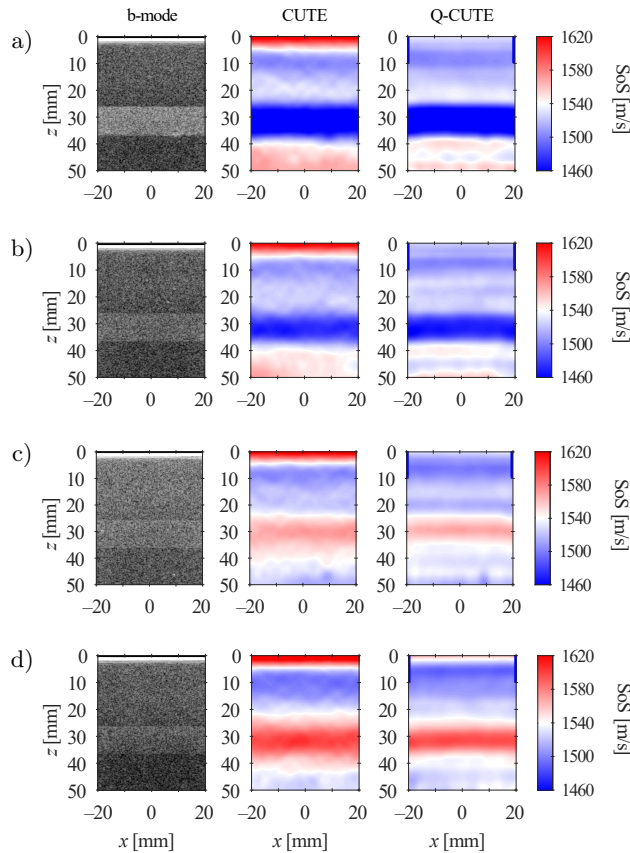
Fig. 6. Images of layers: b-mode (left), SoS images obtained with the use of CUTE (middle), and with Q-CUTE (right). SoS in the layers is: a) 1480 m/s; b) 1510 m/s; c) 1570 m/s; d) 1600 m/s.
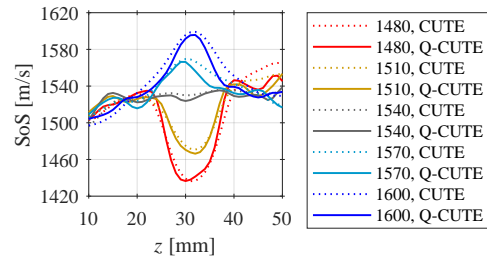

Fig. 7. SoS z-profiles for the homogeneous and layer cases for CUTE (dotted lines) and Q-CUTE (solid lines).
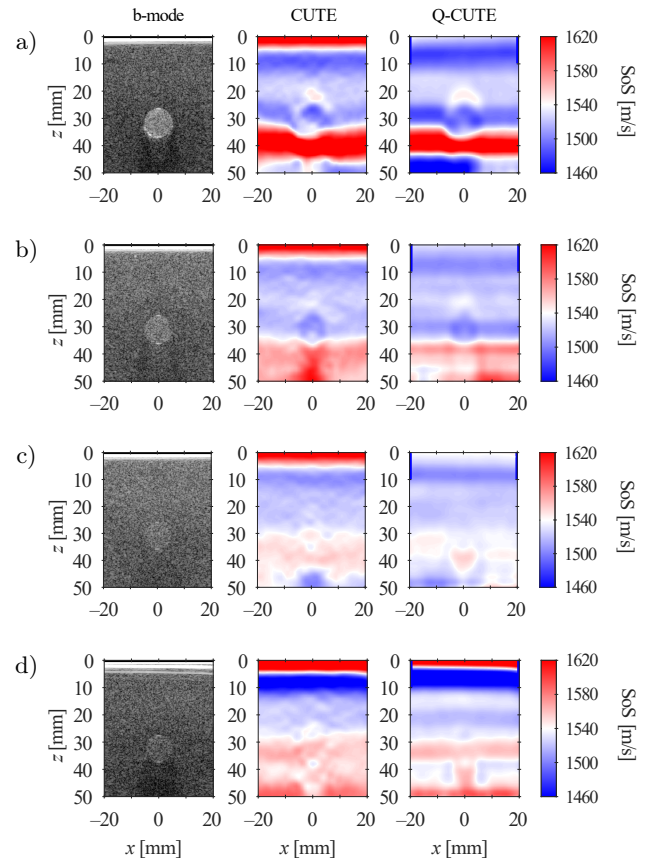

Fig. 8. Images of circular objects: b-mode (left), SoS images obtained with the use of CUTE (middle), and with Q-CUTE (right). SoS in the objects is: a) 1480 m/s; b) 1510 m/s; c) 1570 m/s; d) 1600 m/s.

presents the case of a homogeneous SoS spatial distribution, while layer cases are shown in Fig. 6, and circular object cases are presented in Fig. 8. For a better assessment of the layer images, Fig. 7 shows the averaged z-profiles of the SoS maps shown in Fig. 6.

In the cases shown in Fig. 8, both SoS imaging algorithms fail to reveal the presence of circular objects. To better understand the reasons behind these results, let us analyze the $\Delta\tau$ signal. Each row in Fig. 9 contains $\Delta\tau$ maps for a few angle pairs $[\theta_m, \theta_n]$. The first row (Fig. 9a) shows the same measurement case for which the results in Fig. 8 were computed. The fol-

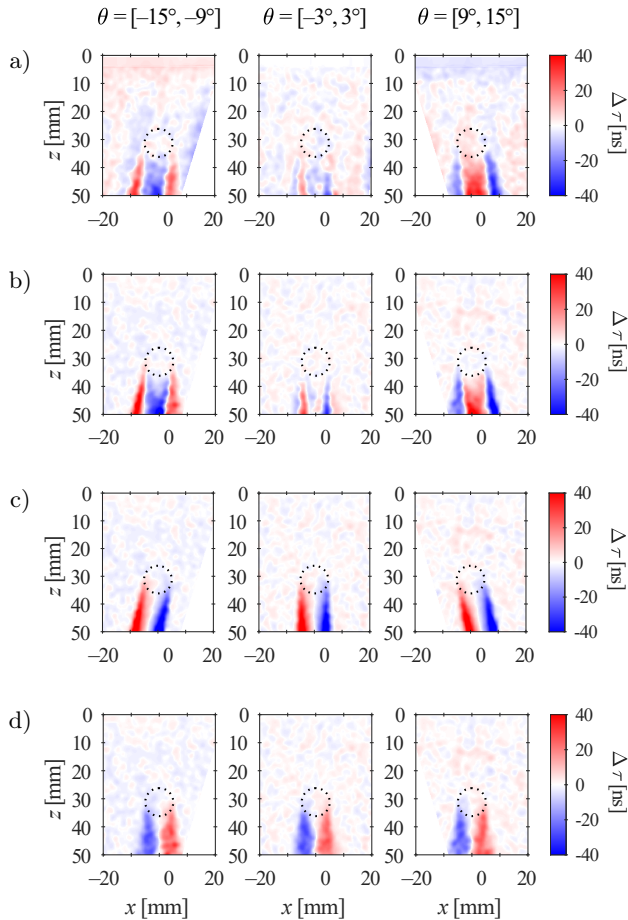$\theta = [-15°, -9°] \qquad \theta = [-3°, 3°] \qquad \theta = [9°, 15°]$

Fig. 9. Images of $\Delta\tau$ for a circular object with an SoS of 1570 m/s for a) measurement data, and for simulated data with the object present in: b) both transmit and receive paths; c) transmit path only; d) receive path only.
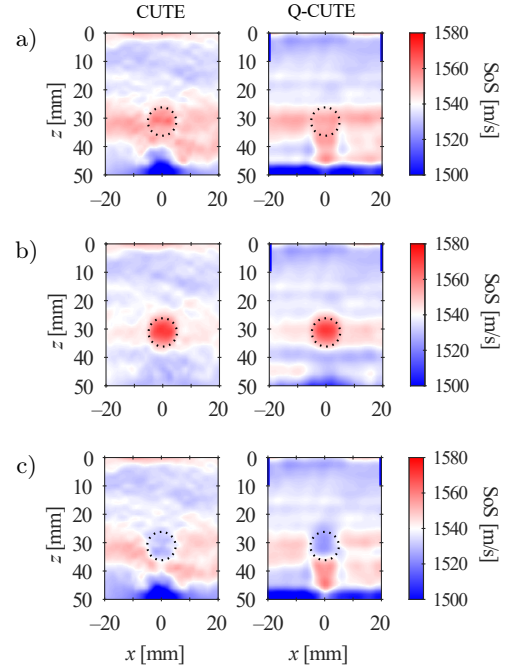

Fig. 10. Images of SoS obtained with the use of CUTE (left) and Q-CUTE (right) algorithms from simulation data. The SoS map used in simulations contained a circular object with an SoS of 1570 m/s present in: a) both transmit and receive paths; b) transmit path only; c) receive path only. The object position is marked with a dashed line.
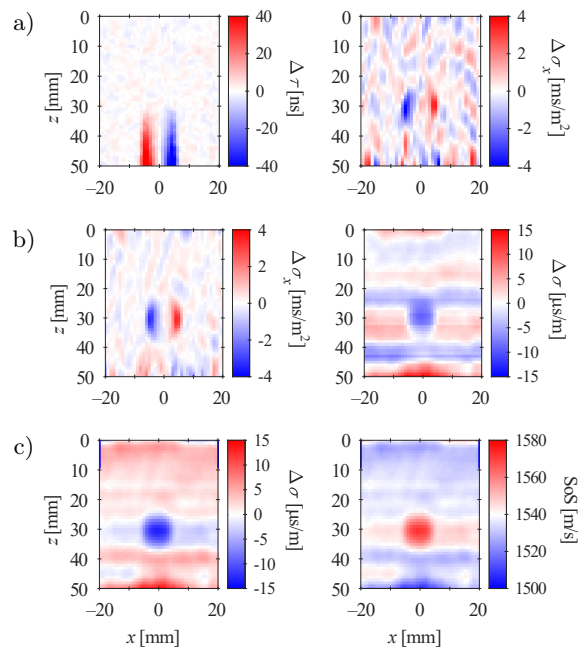

Fig. 11. Images of the signal at selected stages of Q-CUTE processing: a) $\Delta\tau$ (−3°, 3°); b) $\Delta\sigma_x$ (−3°, 3°); c) $\overline{\Delta\sigma_x}$; d) $\Delta\sigma$ without C correction; e) $\Delta\sigma$ with C correction; f) final SoS estimate.

lowing rows (Figs. 9b–9d) present $\Delta\tau$ maps for the simulated data. In the second row (Fig. 9b), the simulated SoS map was the same as in the measurement case, i.e., it contained a circular object with an SoS of 1570 m/s. Subsequently, the presence of the object was limited to the transmit propagation paths only, which led to $\Delta\tau$ maps shown in the third row (Fig. 9c). Analogously, the fourth row (Fig. 9d) illustrates the case with the object being present along the receive propagation paths only.

There is a similarity between the $\Delta\tau$ maps obtained for the measurement (Fig. 9a) and simulation (Fig. 9b). In both cases, they contain a strong component that changes its orientation and a weaker, stationary component in the background. These components are well separated in Figs. 9c and 9d.

The $\Delta\tau$ maps obtained through the simulations (Figs. 9b–9d) were subjected to further processing, resulting in the final SoS images shown in Fig. 10. The signal at intermediate processing steps in the Q-CUTE algorithm for the $\Delta\tau$ data as in Fig. 9c is presented in Fig. 11.

4. Discussion

The presented results allow an evaluation and comparison of the Q-CUTE algorithm with the spatial do-

main CUTE method in terms of SoS imaging quality. As shown in the homogeneous medium case (Fig. 5), both algorithms produce similar artifacts in the shallow area (up to approximately 15 mm depth). Moreover, both methods tend to underestimate the global SoS. Moving on to the layer cases (Figs. 6, 7), the imaging of horizontal objects shows similar dynamics, and the overall underestimation is visible here as well. Further problems appear when a non-layered SoS case is considered. The circular inclusion was not exposed by any of the algorithms (Fig. 8).

The reason for this is the fact that in the $\Delta\tau$ signal, there is a component (Fig. 9d) related to the receive propagation paths passing through the circular object. This component was assumed to be zero and was therefore neglected in the mathematical model. However, as shown through simulations (Fig. 9), it significantly contributes to the $\Delta\tau$ signal. Unfortunately, this contribution is opposite to the useful part of the signal related to the circular object. As a result, imaging of local, non-layered inclusions with CUTE algorithms based on Eq. (4) becomes difficult (Fig. 10).

This problem was already identified in (Stähli *et al.*, 2020). The authors proposed to abandon the assumption of canceling the return path influence. They introduced a new version of the CUTE algorithm that takes into account the receive paths. In the computation of the $\Delta\tau$ signal, the algorithm requires that the transmit-receive mid-angle in the compared CRF images remains the same.

The Q-CUTE algorithm cannot be easily adapted to this new approach. Representing four propagation paths (transmit and receive paths for two acquisitions) with a single common integration path would impose narrow limits on transmit and receive angles. Moreover, at least a quadratic approximation would be required in place of Eq. (9), which could cause further complications.

A more promising way might be to separate or filter out the unwanted receive component. One of the options is to replace the $\Delta\tau$ with differences between $\Delta\tau$ obtained for different sets of θ angles. This approach would eliminate the stationary receive component. This, of course, would require major modifications to the algorithm to account for the change in the $\Delta\tau$ signal definition.

Having discussed the quality of the SoS imaging, it is also worth mentioning other features of the Q-CUTE algorithm. One of its advantages is the modular structure. It makes it possible to control the data at each stage of the processing (Fig. 11), helping to understand the signal and allowing for easy optimization of the algorithm. By contrast, in the case of the CUTE method, a deeper analysis of the signal and its processing is difficult as the algorithm is not modular and is predetermined by the matrix inversion process. One can analyze the impulse responses included in the inverse transformation matrix, but without descriptions like in Eq. (12) it is hard to understand them. There are also other approaches to SoS imaging that hold promise but provide little in terms of understanding the signal. This especially applies to deep learning methods, which have become popular in many areas and have also been used in SoS imaging (Feigin *et al.*, 2020; Young *et al.*, 2022). Their cognitive value, however, is limited to the so-called maps of attention.

Another aspect is the computational complexity. As described in the Methods section, the computational complexity of Q-CUTE is $O(n)$ and, in most cases, it will be significantly lower than the complexity of the spatial domain CUTE which is $O(n^2)$. In practice, it translates into shorter execution times, which was observed when generating the images presented in the Results section. The algorithms were run in MATLAB on a PC with an Intel Core i7-6900K CPU. The common preprocessing took 340 ms, while the remaining parts of the algorithms took 30 ms and 310 ms for Q-CUTE and CUTE, respectively. Consequently, Q-CUTE ran nearly twice as fast as CUTE (2.7 fps versus 1.5 fps). This was the case with the common preprocessing designed with a focus on high-quality $\Delta\tau$ signal. A more balanced quality/performance ratio would lead to overall higher frame rates with a larger difference between Q-CUTE and CUTE. Obviously, single frames per second cannot be considered a satisfactory result in the target applications, but this can be improved for both algorithms through balanced preprocessing, optimized implementations, and of course, the usage of more powerful hardware. In the case of large US devices with high computing power, the computational complexity of the algorithms may not be a decisive factor. However, there is an emerging market of portable medical devices with their limitations. They are battery-powered and are enclosed in tight housing, impeding heat dissipation. For these reasons, power consumption has to be kept at relatively low level. This, in turn, puts constraints on the hardware, limiting its computing power. In this type of US devices, the algorithms such as Q-CUTE can find their use.

5. Conclusions

In this paper, an algorithm for SoS imaging in conventional ultrasound sonography was presented. The algorithm, called Q-CUTE, is characterized by low computational complexity, which allows implementing it on US devices with hardware limitations, e.g., on portable US devices. Its modular structure makes it possible to analyze the data at each stage, helping to understand the signal and optimize the processing.

The Q-CUTE algorithm performs on a similar level as the reference CUTE algorithm. It allows for imaging

the horizontally layered structures, a frequent case in US. These structures include tissues such as skin, fat, muscles, etc., forming layers of different SoS. Therefore, the Q-CUTE algorithm might be used for determining SoS profiles for aberration correction in conventional US.

However, problems arise when revealing non-layered structures, such as circular inclusions. To overcome them, a new approach to the $\Delta\tau$ computation will be analyzed in the future research. This will require further modifications to the mathematical model and the implementation of the Q-CUTE algorithm.

Acknowledgments

References

1. ANDRÉ M., WISKIN J., BORUP D., JOHNSON S., OJEDA-FOURNIER H., OLSON L. (2012), Quantitative volumetric breast imaging with 3D inverse scatter computed tomography, [in:] *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, **34**: 1110–1113, doi: 10.1109/embc.2012.6346129.

2. CACKO D., LEWANDOWSKI M. (2022), Shear wave elastography implementation on a portable research ultrasound system: Initial results, *Applied Sciences*, **12**(12): 6210, doi: 10.3390/app12126210.

3. CHANG S.H., PARK S.B., CHO G.H. (1993), Phase-error-free quadrature sampling technique in the ultrasonic B-scan imaging system and its application to the synthetic focusing system, *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, **40**(3): 216–223, doi: 10.1109/58.216834.

4. COBBOLD R.S.C. (2007), *Foundations of Biomedical Ultrasound*, Oxford University Press.

5. FEIGIN M., FREEDMAN D., ANTHONY B.W. (2020), A deep learning framework for single-sided sound speed inversion in medical ultrasound, *IEEE Transactions on Biomedical Engineering*, **67**(4): 1142–1151, doi: 10.1109/tbme.2019.2931195.

6. Foundation for Research on Information Technologies in Society (n.d.), *Speed of Sound*, https://itis.swiss/virtual-population/tissue-properties/database/acoustic-properties/speed-of-sound/ (access: 12.10.2023).

7. GHOSHAL G., LAVARELLO R.J., KEMMERER J.P., MILLER R.J., OELZE M.L. (2012), Ex vivo study of quantitative ultrasound parameters in fatty rabbit livers, *Ultrasound in Medicine and Biology*, **38**(12): 2238–2248, doi: 10.1016/j.ultrasmedbio.2012.08.010.

8. JAEGER M., HELD G., PREISSER S., PEETERS S., GRÜNIG M., FRENZ M. (2014), Computed ultrasound tomography in echo mode (CUTE) of speed of sound for diagnosis and for aberration correction in pulse-echo sonography, [in:] *Proceedings of SPIE 9040, Medical Imaging 2014: Ultrasonic Imaging and Tomography*, **9040**: 90400A, doi: 10.1117/12.2042993.

9. JAEGER M., HELD G., PEETERS S., PREISSER S., GRÜNIG M., FRENZ M. (2015), Computed ultrasound tomography in echo mode for imaging speed of sound using pulse-echo sonography: Proof of principle, *Ultrasound in Medicine and Biology*, **41**(1): 235–250, doi: 10.1016/j.ultrasmedbio.2014.05.019.

10. JAEGER M., FRENZ M. (2015), Quantitative imaging of speed of sound in echo ultrasonography, [in:] *IEEE International Ultrasound Symposium*, https://www.youtube.com/watch?v=Ck75XbfLQtY (access: 12.10.2023).

11. JENSEN J.A. (1996), Field: A program for simulating ultrasound systems, [in:] *Medical & Biological Engineering & Computing*, **34**(1): 351–353.

12. KARWAT P. (2019), Computationally efficient algorithm for sound speed imaging in pulse-echo ultrasound, *Proceedings of Meetings on Acoustics*, **38**(1): 020005, doi: 10.1121/2.0001109.

13. LYONS R.G. (2004), *Understanding Digital Signal Processing,* 2nd ed., Prentice Hall.

14. SANABRIA S.J., OZKAN E., ROMINGER M., GOKSEL O. (2018), Spatial domain reconstruction for imaging speed-of-sound with pulse-echo ultrasound: Simulation and in vivo study, *Physics in Medicine and Biology*, **63**(21): 215015, doi: 10.1088/1361-6560/aae2fb.

15. STÄHLI P., KURIAKOSE M., FRENZ M., JAEGER M. (2020), Improved forward model for quantitative pulse-echo speed-of-sound imaging, *Ultrasonics*, **108**: 106168, doi: 10.1016/j.ultras.2020.106168.

16. TREEBY B.E., COX B.T. (2010), k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields, *Journal of Biomedical Optics*, **15**(2): 021314, doi: 10.1117/1.3360308.

17. YOUNG J.R., SCHOEN S., KUMAR V., THOMENIUS K., SAMIR A.E. (2022), SoundAI: Improved imaging with learned sound speed maps, [in:] *2022 IEEE International Ultrasonics Symposium*, doi: 10.1109/IUS54386.2022.9958284.