Journal of Real-Time Image Processing manuscript No.
(will be inserted by the editor)

Acceleration framework with GPU for minimum variance adaptive beamforming algorithms in medical ultrasonic imaging

Junying Chen[a,1], **Jinhui Chen**[b,1], **Huaqing Min**[c,1]

[1]Guangzhou Key Laboratory of Robotics and Intelligent Software, School of Software Engineering, South China University of Technology, Guangzhou 510006, China

Received: date / Accepted: date

Abstract Minimum variance adaptive beamforming algorithms have been studied in medical ultrasonic imaging during the past decade. These algorithms are proposed to improve the ultrasonic image quality, including lateral resolution and contrast. However, the image quality is improved at the expense of imaging speed. It takes longer time to compute a higher-quality image. On the other hand, GPU has been widely used in image and signal processing, acceleration of computer graphics and so on because of its powerful parallel computing capacity. Hence, while implementing minimum variance adaptive beamforming algorithms in real-time, GPU is usually conducted. Furthermore, in order to conduct a proper parallel computing scheme, an acceleration framework for various minimum variance adaptive beamforming algorithms is necessary. The paper introduce an acceleration framework with GPU for minimum variance adaptive beamforming algorithms to help achieve real-time performance. This framework divides the algorithm into three stages and use GPU as much as possible for parallel computing in each stage. While implementing various minimum variance beamforming algorithms in real-time, trade-offs between image quality and imaging speed are usually conducted too. Experiments are carried out on several algorithms in this work and it provides an image quality growth trend for a given algorithm, so that the image quality acceptable region can be recognized, which helps to conduct an appropriate real-time implementation trade-off.

Keywords Acceleration framework · GPU · Real-time performance · Adaptive beamforming algorithm · Medical ultrasonic imaging

[a]e-mail: jychense@scut.edu.cn
[b]e-mail: chen.jinhui@mail.scut.edu.cn
[c]e-mail: hqmin@scut.edu.cn

1 Introduction

In the field of medical ultrasonic imaging, delay-and-sum (DAS) beamforming [7] is still extensively used nowadays in clinical applications, because of its real-time capabilities. However, the image quality of DAS beamforming algorithm cannot provide enough anatomical details when encountering some complicated diagnostic cases. Therefore, the more high-definition beamforming algorithm has gradually become a hot topic in the field of ultrasonic imaging. Minimum variance (MV) adaptive beamforming was first introduced by Capon in 1969 [4] and was applied to medical ultrasonic imaging by Synnevag in 2007 [20]. MV adaptive beamforming could get better image quality because it computes the apodization weights dynamically during run-time in response to the input echo data, which could obtain the real-time statistical characteristics of the signal.

During the past decade, many MV adaptive beamforming algorithms have been developed to keep improving the ultrasonic image quality. As reported in the related literature, advanced MV adaptive beamforming algorithms can improve the lateral resolution as well as contrast of the ultrasonic images. Besides, MV adaptive beamforming algorithms also concerned about the robustness enhancement of the beamformer [10]. In the development of MV adaptive beamforming, iterative MV (IMV) beamforming algorithm was also proposed to enhance the robustness of MV beamformer, which output a high signal-to-interference-plus-noise ratio [16]. Moreover, A forward-backward MV (FBMV) beamforming algorithm was introduced to enhance the robustness of MV beamforming and improve the image contrast of the output images, without compromising the resolution of basic MV beamforming algorithm [2]. Furthermore, coherence factor(CF) is introduced into MV algorithms to improve ultrasonic image quality [3, 17], such as MV beamforming was combined with general coherence factor (GCF)

method [18, 22]. On the other hand, eigenspace-based MV (ESBMV) beamforming algorithm was applied to medical ultrasonic imaging to enhance the image quality of MV beamforming algorithm [1]. Based on ESBMV, it combined with sign coherence factor (SCF) method was raised to improve the lateral resolution, as well as contrast and robustness [12]. Also, the combination of ESBMV and GCF [6] and the combination of ESBMV and correlation coeficient eigenvalues weighting [24] were used to further improve the quality of MV beamformed images. And in 2016, ESBMV was combines with a subarray coherence based postfilter, for improving the quality of ultrasound plane-wave imaging [23]. In 2017, Ziksari et al. studied integrating MV beamformer and correction algorithm which confirmed that the correction method retrieves the efficient performance of MV [25].

Moreover, MV adaptive beamforming is gradually being applied to other fields, like photoacoustic imaging. In 2018, a novel beamformer using MV adaptive beamforming combined with Delay-Multiply-and-Sum (DMAS), which was introduced having lower sidelobes comparedto DAS, is introduced by Moein et al. to improves photoacoustic imaging qualit [14]. Later, another beamformer using the ESBMV method combined with DMAS was introduced to further improves the imaging qualit [15].

Real-time imaging is the critical characteristic of medical ultrasonic imaging. An algorithm should be able to run in real-time imaging so as to apply it to the real clinical diagnoses. Hence, not only the image quality is important for an MV adaptive beamforming algorithm, but also the imaging speed is crucial to the MV adaptive beamforming algorithm. But image quality and imaging speed are usually two contradictory performance indicators for medical ultrasonic imaging. It usually takes longer time to compute a higher-quality image with MV adaptive beamforming algorithm [19].

There are some ways to improve real-time performance in terms of signal reception and transmission, like Multi-Line Acquisition (MLA) and Multi-Line Transmission (MLT) [13]. On the other hand, GPU has been widely used in image and signal processing, acceleration of computer graphics and so on because of its powerful parallel computing capacity. Hence, while implementing minimum variance adaptive beamforming algorithms in real-time, GPU is usually conducted. Furthermore, in order to conduct a proper parallel computing scheme, an acceleration framework for various minimum variance adaptive beamforming algorithms is necessary. The paper introduce an acceleration framework with GPU for minimum variance adaptive beamforming algorithms to help achieve real-time performance. This framework divides the algorithm into three stages and use GPU as much as possible for parallel computing in each stage. The trade-offs between image quality and imaging speed are often conducted for real-time medical ultrasonic imaging algorithm implementations too. Consequently, the image quality quantitative study of MV adaptive beamforming algorithms is needed, which is supposed to help conducting an appropriate implementation trade-off and help analyzing the performance of the MV beamforming algorithms. In this work, we will carry out experiments on several MV adaptive beamforming algorithms and provides an image quality growth trend for a given algorithm, so that the image quality acceptable region can be recognized, which helps to conduct an appropriate real-time implementation trade-off.

The remaining of the paper is organized as follows. Section 2 briefly analyze the MV adaptive beamforming algorithms, especially their major modifications and improvements over basic MV beamforming algorithm. Section. 3 introduce the acceleration framework with GPU for MV adaptive beamforming algorithms and the detailed parallel accelerated implementation of the algorithms on GPU is described in this section. Experimentations for the algorithms are carried out in Sect. 4 and it will be divided into 2 parts, the first part analyzes the imaging quality, and the second part analyzes the real-time performance. Finally, Sect. 5 draws conclusions.

2 Analysis of MV adaptive beamforming algorithms

Many MV adaptive beamforming algorithms have been developed to keep improving the ultrasonic image quality during the past decade. This section start from the regular MV beamforming as a basic MV beamforming algorithm and then analyzes six other algorithms as expansion cases.

2.1 Basic MV beamforming

In basic MV beamforming algorithm [4, 20], the beamformer input is a data matrix obtained from M ultrasonic transducer elements. A data array $\boldsymbol{x}$ consisting of M input data samples is utilized as the input for one pixel value calculation. Each of the M data samples comes from each transducer element and is time delay compensated. Sub-array averaging is applied subsequently, whose size is L. Hence, $(M-L+1)$ sub-arrays are constructed from $\boldsymbol{x}$. Then, the covariance matrix $\boldsymbol{R}_{MV}$ is:

$$\boldsymbol{R}_{MV} = \frac{1}{M-L+1}\sum_{i=0}^{M-L}\boldsymbol{x}_i \cdot \boldsymbol{x}_i^H \tag{1}$$

where $\boldsymbol{x}_i$ is the i^{th} sub-array containing i^{th} to $(i+L-1)^{th}$ data samples in $\boldsymbol{x}$. The adaptive weight vector is:

$$\boldsymbol{w}_{MV} = \frac{\boldsymbol{R}_{MV}^{-1}\cdot\boldsymbol{a}}{\boldsymbol{a}^H\cdot\boldsymbol{R}_{MV}^{-1}\cdot\boldsymbol{a}} \tag{2}$$

where steering vector $\boldsymbol{a}$ is all ones. Hence, the pixel value is:

$$v_{MV} = \frac{1}{M-L+1} \sum_{i=0}^{M-L} \boldsymbol{w}^H \cdot \boldsymbol{x}_i \tag{3}$$

2.2 MV beamforming with diagonal loading

Because of the inaccurate estimation of frequency or arrival direction, the high-quality imaging performance of basic MV algorithm declines rapidly. In order to enhance the robustness of basic MV algorithm, diagonal loading (DL) is applied [10, 21]. The method adds a certain proportion of unit matrix to $\boldsymbol{R}_{MV}$:

$$\boldsymbol{R}_{MV\_DL} = \boldsymbol{R}_{MV} + \delta \cdot trace(\boldsymbol{R}_{MV}) \cdot \boldsymbol{I} \tag{4}$$

where δ is a diagonal loading factor and $\boldsymbol{I}$ is a unit matrix. Then replace $\boldsymbol{R}_{MV}$ by $\boldsymbol{R}_{MV\_DL}$ in (2) to further calculate v_{MV_DL}. Because diagonal loading principle adds a certain proportion of Gaussian white noise to original signals, so as to increase the independence between the signals, it causes some loss of lateral resolution. In order not to lose much resolution, δ should not be too large, which is usually $1/(100L)$.

2.3 Iterative MV beamforming

IMV beamforming algorithm was proposed to improve the algorithm accuracy and robustness using noise covariance matrix in a number of iterations [16]. In this algorithm, v_{MV} obtained in (3) is taken as the estimate of expected signal amplitude $s_{IMV} = v_{MV}$. Then s_{IMV} is separated from the delayed input data, leaving the remainder being the interference noise component:

$$\boldsymbol{x}_{i\_IMV} = \boldsymbol{x}_i - s_{IMV} \cdot \boldsymbol{a} \tag{5}$$

Replace $\boldsymbol{x}_i$ by $\boldsymbol{x}_{i\_IMV}$ and calculate from (1) to (3) to get v_{IMV}, then set $s_{IMV} = v_{IMV}$ to repeat optimization iteration.

2.4 MV beamforming with forward-backward averaging

FBMV beamforming algorithm [2, 9] was used to improve the contrast and robustness of basic MV algorithm. FBMV algorithm estimates a more accurate covariance matrix with forward-backward spatial averaging, instead of conventional forward-only spatial averaging. $\boldsymbol{R}_{MV}$ in (1) is referred to as the forward estimate $\boldsymbol{R}_F = \boldsymbol{R}_{MV}$ to obtain the backward estimate $\boldsymbol{R}_B = \boldsymbol{J} \cdot \boldsymbol{R}_F^T \cdot \boldsymbol{J}$, where $\boldsymbol{J}$ is a reversal matrix. Subsequently, the forward-backward averaging estimate is:

$$\boldsymbol{R}_{FB} = \frac{1}{2}(\boldsymbol{R}_F + \boldsymbol{R}_B) \tag{6}$$

$\boldsymbol{R}_{FB}$ replaces $\boldsymbol{R}_{MV}$ in (2) to calculate adaptive weight vector.

2.5 MV beamforming with GCF

In order to obtain high resolution and correct phase errors induced by velocity inhomogeneity of tissue, MV beamforming with GCF [11] was raised [22]. The array data is first transformed from array space to beam space by Fourier transform:

$$b(k) = e^{j\pi k} \sum_{i=0}^{M-1} x_i(t) \cdot e^{-j2\pi t \frac{k}{M}} \tag{7}$$

where $b(k)$ is spatial frequency data in beam space, and $x_i(t)$ is i^{th} element in $\boldsymbol{x}$ at sampling time t. So, GCF at time t is:

$$GCF(t) = \frac{\sum_{k\in(0,1,\ldots,K)} |b(k)|^2}{\sum_{k=0}^{M-1} |b(k)|^2} \tag{8}$$

where K is the energy ratio controlling the low frequency component of $GCF(t)$, which can be changed to adjust the algorithm performance. Finally, the amplitude estimate is: $v_{MV\_GCF} = GCF(t) \cdot v_{MV}$

2.6 Eigenspace-based MV beamforming

ESBMV beamforming algorithm was proposed to significantly enhance the ultrasonic image contrast [1, 12]. ESBMV algorithm utilizes the eigenstructure of $\boldsymbol{R}_{MV}$ to suppress off-axis signals while keeping on-axis ones. The eigendecomposition of $\boldsymbol{R}_{MV}$ is:

$$\boldsymbol{R}_{MV} = \boldsymbol{U} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{U}^H = \boldsymbol{U}_S \cdot \boldsymbol{\Lambda}_S \cdot \boldsymbol{U}_S^H + \boldsymbol{U}_N \cdot \boldsymbol{\Lambda}_N \cdot \boldsymbol{U}_N^H \tag{9}$$

where $\boldsymbol{\Lambda} = diag(\lambda_0, \lambda_1, \ldots, \lambda_{L-1})$ in which $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{L-1}$, and $\boldsymbol{U} = [\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{L-1}]$ in which $\boldsymbol{v}_i$ is the eigenvector corresponding to λ_i. $\boldsymbol{U}_S = [\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{N-1}]$ is signal subspace consisting of N eigenvectors corresponding to larger eigenvalues, while $\boldsymbol{U}_N = [\boldsymbol{v}_N, \boldsymbol{v}_{N+1}, \ldots, \boldsymbol{v}_{L-1}]$ ifs noise subspace. The value of N determines the capability of keeping main-lobe signal and reducing side-lobe signals. Its value is usually larger than α times maximum eigenvalue ($0.1 \leq \alpha \leq 0.5$) or β times minimum eigenvalue ($10 \leq \beta \leq 50$). Hence, the weight vector is: $\boldsymbol{w}_{ESBMV} = \boldsymbol{U}_S \cdot \boldsymbol{U}_S^H \cdot \boldsymbol{w}_{MV}$.

2.7 Eigenspace-based MV beamforming with SCF

In order to further improve image resolution of ESBMV algorithm, SCF was introduced to eigenspace [12]. SCF at sampling time t is:

$$SCF(t) = \left| 1 - \sqrt{1 - \left(\frac{1}{M} \sum_{i=0}^{M-1} c_i(t) \right)^2} \right|^q \tag{10}$$

where $q \geq 0$ is used to adjust SCF sensitivity. As q increases, the algorithm can better suppress side-lobe amplitude and reduce main-lobe width. $c_i(t)$ is sign value of related data:

$$c_i(t) = \begin{cases} +1, & x_i(t) \geq 0 \\ -1, & x_i(t) < 0 \end{cases} \tag{11}$$

Its amplitude estimate is: $v_{ESBMV\_SCF} = SCF(t) \cdot v_{ESBMV}$.

3 Acceleration framework with GPU

Although MV adaptive beamforming algorithm outputs high-quality medical ultrasonic images, it is computationally demanding. The high computational complexity of MV adaptive beamforming algorithm hinders MV algorithm being implemented in real-time on conventional computing platforms, such as conventional ARM processors. Therefore, the implementation of MV adaptive beamforming algorithm on computing platform with high-performance GPU is explored in this paper, so as to produce an acceleration framework wiht GPUfor MV adaptive beamforming algorithmto improve their real-time imaging capability.

The input data delay calculation and the covariance matrix calculation consumed most of the execution time of the sequential MV code. As L/M increased, the percentage of execution time used for covariance matrix calculation and adaptive weight calculation increased, but the percentage of execution time for input data delay calculation decreased. Such changes conformed to the computational complexities of different calculation steps. For example, the input data delay calculation had a computational complexity of $O(n)$, the covariance matrix calculation had a computational complexity of $O(n^2)$, and the adaptive weight calculation had a computational complexity of $O(n^3)$. Based on the analysis of the sequential MV code, the code parts with high computational complexity or high percentage of execution time should be implemented on the GPU platform. As a result, also considering reducing the hardware/software communications between GPU and CPU, all parts of the MV adaptive beamforming algorithm were decided to implement on the GPU platform.

3.1 Overall design overview

Figure 1 show the overall design framework of implementation for MV adaptive beamforming. This section will introduce the implementation strategies on GPU.

3.1.1 Resource allocation policy

In order to effectively utilize the desktop GPU to implement the MV adaptive beamforming algorithm efficiently, two implementation strategies are basically applied.

The first implementation strategy is about the allocation of the GPU computing resources. In the medical ultrasonic image formation process, the image pixel amplitude estimate values of the whole image are calculated following MV adaptive beamforming algorithm calculation steps as described in Sect. 2. The image pixels are organized in rows and columns, which can be mapped to the GPU compute grid with a two-dimensional GPU compute block allocation. As a result, each GPU compute block is responsible to the computation of one image pixel amplitude estimate value calculation. The program steps used to calculate the image pixel amplitude estimate value are executed inside one block via the simultaneous computation of the parallel threads within the block. Finally, the sequential operations of the program, which cannot be executed at the same time, are computed inside the threads. The best practices of the GPU block size and the GPU thread size rely on the arrangement of the computational resources on the computing platform according to the computational problem size.

In addition to the allocation of GPU computing resources, the memory access strategy of GPU implementation has a important impact on the overall GPU computational speed. In a GPU program memory assignment, small-size and frequently-used variables can be stored in the register files, while large amount of data should be stored in the global memory. Furthermore, The shared memory is usually used for the shared data space utilized in the computation among the GPU compute threads within a specific GPU compute block.

3.1.2 Parallel implementation framework for processes

Referring to the allocation of GPU computing resources illustrated as above, the high-level design block diagram of the MV adaptive beamforming algorithm for GPU implementation is demonstrated in Fig. 1. Based on the overall design block diagram, the fine-grained parallelization of the algorithm implementation was conducted on the desktop GPU.

In the design, the MV beamforming implementation took M receive channels as its inputs to generate an amplitude estimation of one image pixel. The input data from the ultrasonic echo receive channels was also known as pre-beamform data before the beamforming process, and the output pixel value was also known as post-beamform data after the beamforming process.The whole parallel computing process will be divided into the following three steps.

Delay calculation Each receive channel streamed input echo samples to the delay calculation block, forming an $M \times 1$ vector of delayed echo samples as its output. The purpose of the delay calculation block was to align the input ultrasonic echoes based on the delay information among the receive channels. The delayed echo vector must subsequently be multiplied by the adaptive apodization weights.

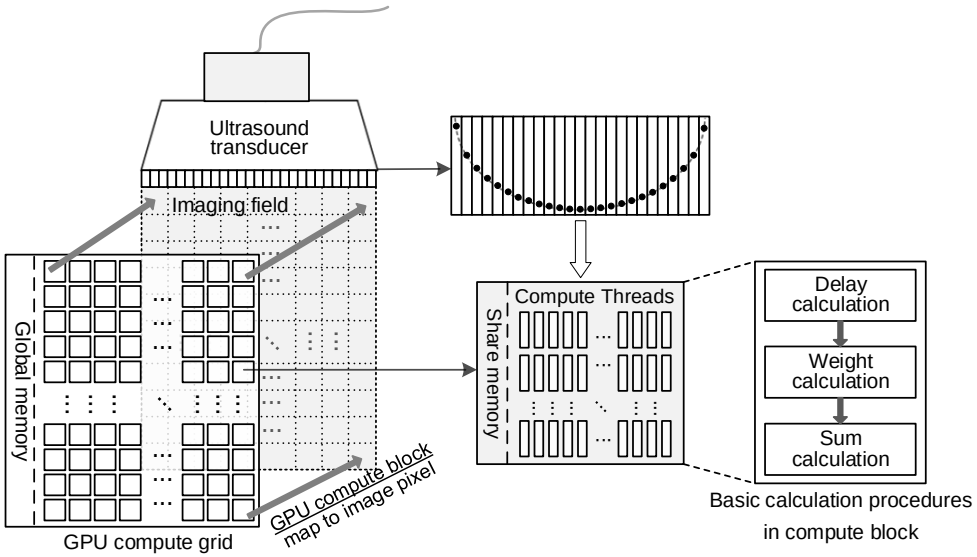

Fig. 1 Overall design framework

Weight calculation After the Delay calculation, the delayed echo vector will be multiplied subsequently by the adaptive apodization weights. Hence, while the adaptive weight calculation block was calculating the adaptive weights, the echo vector was stored in the data storage buffer for later usage, which was built with GPU shared memory. The purpose of data storage buffer was to let the delayed echo vector wait until the adaptive weight calculation block finished its workload.

Sum calculation Finally, the pixel amplitude estimation block output the final pixel value. It first multiplied $(M-L+1)$ segmented delayed echo vectors $\boldsymbol{echo}_{subk}(p_s)(k=0,1,...,M-L)$ and their adaptive weights. The results of these $(M-L+1)$ pixel value estimates were then averaged, and finally add up to obtain the final pixel amplitude value output.

3.2 Implementation for Algorithms

3.2.1 Basic MV beamforming

Many adaptive beamforming algorithms are improved from basic MV beamforming, so the GPU implementation of basic MV beamforming can be used as a basis for other algorithms. This paper introduce an framework from basic MV beamforming as shown in Fig. 2, so that it can serve as a basis for other adaptive beamforming algorithms implementation. In the whole process, the process of delay calculation and sum calculation is relatively simple, while the weight calculation is more complicated. The integration of the mathematical theories into the implementation will be described below.

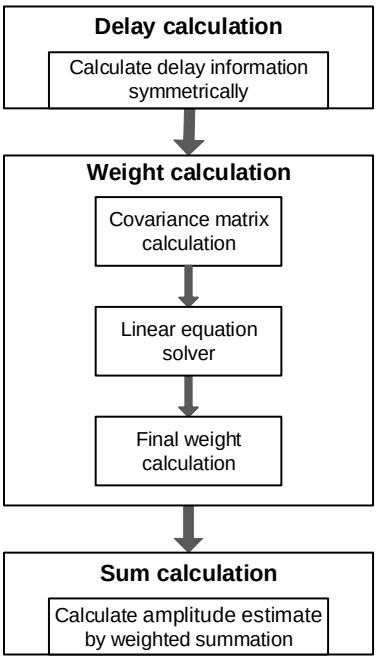

Fig. 2 Computation framework for MV beamforming

The delay calculation of each receive channel can be easily paralleled by multiple threads of calculation block. At this part it is worth mentioning that the receiving element is symmetrical on both sides of the scan line. As shown in Fig. 3, the positions on both sides of SL_k are symmetrical, so the delay information $DL_k - n = DL_k + n$. Therefore, the delay information on one side of the scan line can be calculated and then mapped to the corresponding position on the other side of the scan line, thus reducing the computation by nearly half in this part.

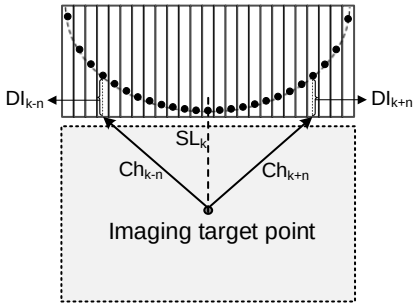

Fig. 3 Receiving element on both sides of the scan line

The weight calculation is the key of algorithm, so the difference of GPU implementation is mainly focused on this part. The adaptive weight calculation block in Fig. 1 performs the core computation of the MV adaptive beamforming algorithm. It consists of three major units: covariance matrix calculation, linear equation solver, and final weight calculation. In order to implement a highly parallelized MV algorithm, probability theories and linear algebra theories were used to optimize the detailed implementation and reduce the computation operations. Here, the inner working principles of these blocks can be refered to [5].

According to (3), it should first calculate the multiplication of weights and delay echo signal vector, and then compute summation for obtaining the final pixel amplitude value. However, this requires a large number of multiplication calculations. This paper extract weights as common factors as:

$$\sum_{i=0}^{M-L}(\boldsymbol{w}^H \times \boldsymbol{x}_i) = \boldsymbol{w}^H \times \sum_{i=0}^{M-L}\boldsymbol{x}_i \tag{12}$$

Thus, we can first sum up the delay echo signal using multi threads and then multiplied by the weights. By this way, this step just neel L multiply operations so that it saves $(M-L)\times L$ multiply operations. Through these parallel computing designs, the powerful computing performance of GPU can be brought into full play, and the fast imaging process of basic MV algorithm can be realized.

3.2.2 MV beamforming with diagonal loading

MV beamforming with diagonal loading mainly adds a certain proportion of white Gaussian noise to the covariance matrix obtained by the basic MV algorithm. Therefore, the parallel implementation scheme of the basic MV algorithm can still be used. On this basis, after the calculation of the covariance matrix is completed, the additional value need be calculated. And then add to the diagonal line of the matrix. This part of the calculation is mainly added to the module of weight calculation as shown in Fig. 4. Because the overlay process has no access conflict, the parallel operation of L threads in the thread block can be carried out to complete the diagonal loading process more quickly and achieve the purpose of fast imaging.

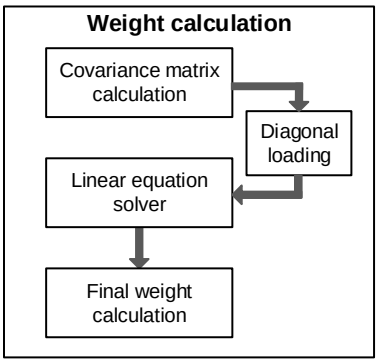

Fig. 4 Diagonal loading in weight calculation

3.2.3 Iterative MV beamforming

The calculation process of the iterative MV algorithm is essentially the same as that of the basic MV algorithm. It just repeats the process of getting the weight in the basic MV algorithm and subtracts the amplitude estimation of the previous iteration. Therefore, by adding a variable to record the median estimate of the previous calculation and subtracting it from the delay signal every time, the pixel amplitude of the iterative MV algorithm can be obtained by further iterative calculation with the parallel computation scheme of the basic MV algorithm. Its framework is shown in Fig. 5. Because of the iterative computation, its imaging speed is dependent on the expected number of iterations compared with the basic MV algorithm.

3.2.4 MV beamforming with forward-backward averaging

It can be seen from the algorithm formula that the MV beamforming with forward-backward averaging corresponds the covariance matrix obtained by the basic MV algorithm to the forward spatial averaging estimation, and the backward spatial averaging value can be obtained in the relative position of the forward estimation. Meanwhile, the covariance matrix of the basic MV algorithm is symmetric. Therefore, after the basic MV algorithm completes the calculation of its covariance matrix, the same parallel scheme can be used to obtain the forward and backward smoothing values respectively, and then combined with the two values to calculate the average value directly to replace the corresponding position in the matrix as shown in Fig. 6. This scheme can achieve good parallel computing without increasing memory overhead, and greatly save the computing time of the joint covariance matrix.

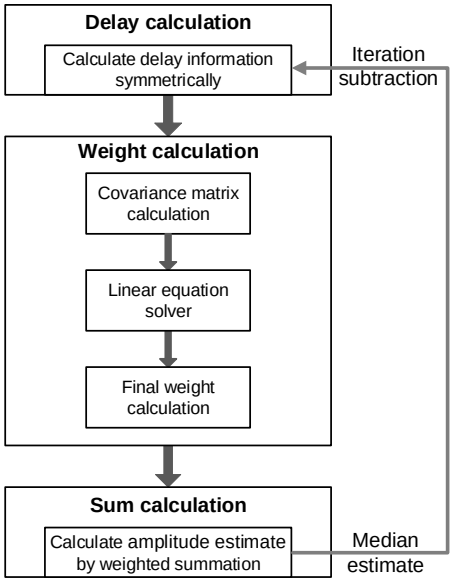

Fig. 5 Framework for Iterative MV beamforming

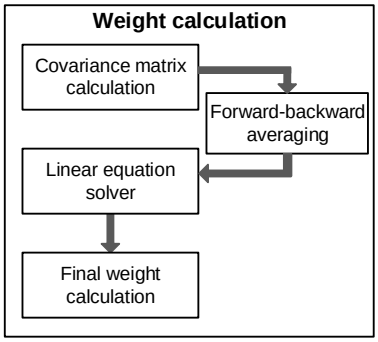

Fig. 6 Forward-backward averaging in weight calculation

3.2.5 MV beamforming with GCF

The MV algorithm with GCF has the same calculation process as the basic MV algorithm. Therefore, the GPU implementation of the basic MV algorithm is still applied in this area. Then, after the pixel amplitude value is obtained, the delay signal data stored in the shared memory is Fourier transformed, and the coherence factor is calculated according to Section 2.5. Finally, the final amplitude value can be obtained by multiplying the final value of basic MV algorithm. Its framework is shown in Fig. 7. Furthermore, cosf, sinf and make_float2 functions can be called to help complete the calculation process. These functions are built-in functions of CUDA, which can be called directly in the kernel function. They are more suitable for running on NVIDIA GPU than custom functions, and have better computational performance.

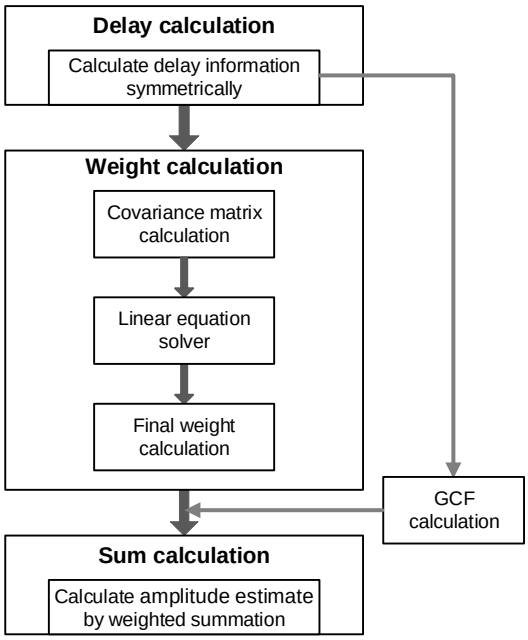

Fig. 7 Framework for MV beamforming with GCF

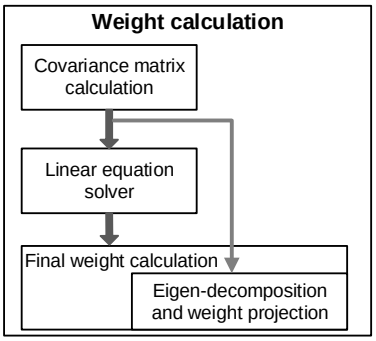

Fig. 8 Weight calculation of eigenspace-based MV beamforming

3.2.6 Eigenspace-based MV beamforming

Eigenspace-based MV beamforming has two more steps, the eigen-decomposition and weight projection, than basic MV algorithm in weight calculation as shown in Fig. 8. For the sake of precision and parallelism, Jocobi decomposition method is used to do eigen-decomposition of covariance matrix, and multi-threads in the thread block are used to implement parallel computing for non-access conflict computing. In order to implement parallel cooperation between threads, it is necessary to add shared memory to the original MV algorithm to store the intermediate values in the process of eigen-decomposition, which is mainly used to store eigenvectors, and the eigenvalues can be stored in the allocated shared memory which is no longer needed in the final computation. The size of shared memory limits the number of GPU compute blocks that can run in parallel in runtime. In order to save memory, the same data with symmetry should be calculated and stored only once in the matrix computing

process, and then read and reuse through the subscript transformation calculation. This not only reduces nearly half of the computation, but also reduces the actual total amount of shared memory and increases the parallel amount of compute blocks at runtime. So as to achieve the purpose of imaging as fast as possible.

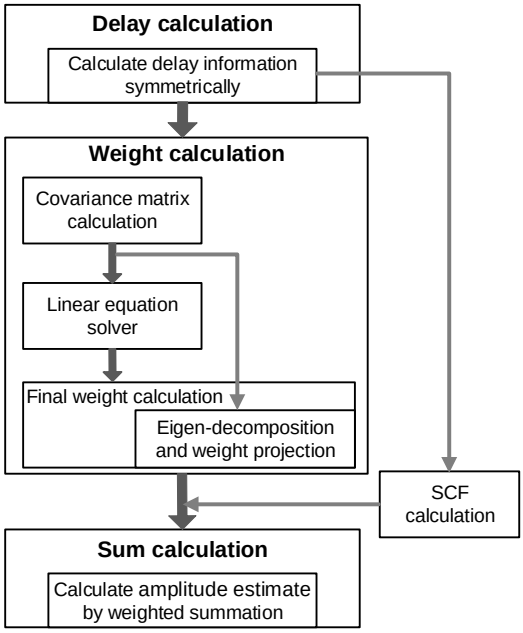

Fig. 9 Framework for eigenspace-based MV beamforming with SCF

3.2.7 Eigenspace-based MV beamforming with SCF

The calculation process of eigenspace-based MV beamforming with SCF is mainly multiplying the amplitude value of eigenspace-based MV beamforming by a sign coherence coefficient. Therefore, its implementation scheme is consistent with eigenspace-based MV beamforming, and only one more step is needed to calculate the value of SCF as coefficient of correction according to (11). Its framework is shown in Fig. 9. Since this algorithm has two parts to improve than the basic MV algorithm, it will take much more computation time.

4 Experiments and Evaluations

4.1 Experimental simulations and setups

In order to evaluate the imaging performance of the algorithms and the real-time performance of the acceleration framework introduced in this paper, we implement the algorithms mentioned above through the acceleration framework, and then statistical various indicators for experimental analysis. The experiments were performed on an Intel Core i7 4790K running at 3.6 GHz and Nvidia GTX 980Ti with 2816 CUDA cores.

Among various image quality evaluation factors, lateral resolution and image contrast are the two most critical evaluation factors. As a result, in order to have an in-depth quantitative study of the image lateral resolution and contrast, simulations were adopted to provide precise imaging scenario settings. The imaging scenarios were simulated using Field II simulator [8]. Simulated points were used to evaluate the quantitative lateral resolution performance of all the MV adaptive beamforming algorithms analyzed in Sect. 2, and simulated cysts were used to evaluate the quantitative contrast performance of all the discussed algorithms.

The simulated single points were located at 25 mm, 40 mm and 55 mm imaging depths respectively in three imaging scenarios, as shown in Fig. 10(a)-(c). For cysts simulation, the centers of simulated cysts were located at 15 mm, 25 mm, 35 mm and 50 mm, and the diameters of the simulated cysts at corresponding imaging depths were 6 mm, 6 mm, 8 mm and 10 mm, as shown in Fig. 11(a). The emission focusing depths of the simulated imaging scenarios were 25 mm, and the simulated ultrasonic transducer had 128 transducer elements with 0.3048 mm element pitch. Besides, the simulated emission center frequency was 5 MHz using a 5 KHz pulse repetition rate. Furthermore, the simulator simulated the imaging scenarios and sampled the received ultrasonic echoes at 40 MHz, which were the input data samples for MV beamformers. The output images of all the MV adaptive beamforming algorithms for the simulated point at 25 mm imaging depth as demonstration examples of simulated single point output images are shown in Fig. 10(d)-(j), and the output images of simulated cysts are shown in Fig. 11(b)-(h).

4.2 Quantitative study of image quality

4.2.1 Quantitative lateral resolution study

In order to measure the lateral resolution of various MV beamforming algorithms quantitatively, the point spread functions of the simulated points shown in Fig. 10(a)-(c) were investigated. The quantitative lateral resolution values at various imaging depths were obtained by calculating the full width at half maximum of the point spread functions at specific imaging depths. The lateral resolution was considered to be better, when the lateral resolution value was smaller. The lateral resolution values at various imaging depths for basic MV beamforming algorithm with various M and L are shown in Fig. 12. The better lateral resolution was obtained with larger M when the imaging depth was closer to the focal depth. Besides, while L/M was growing, the

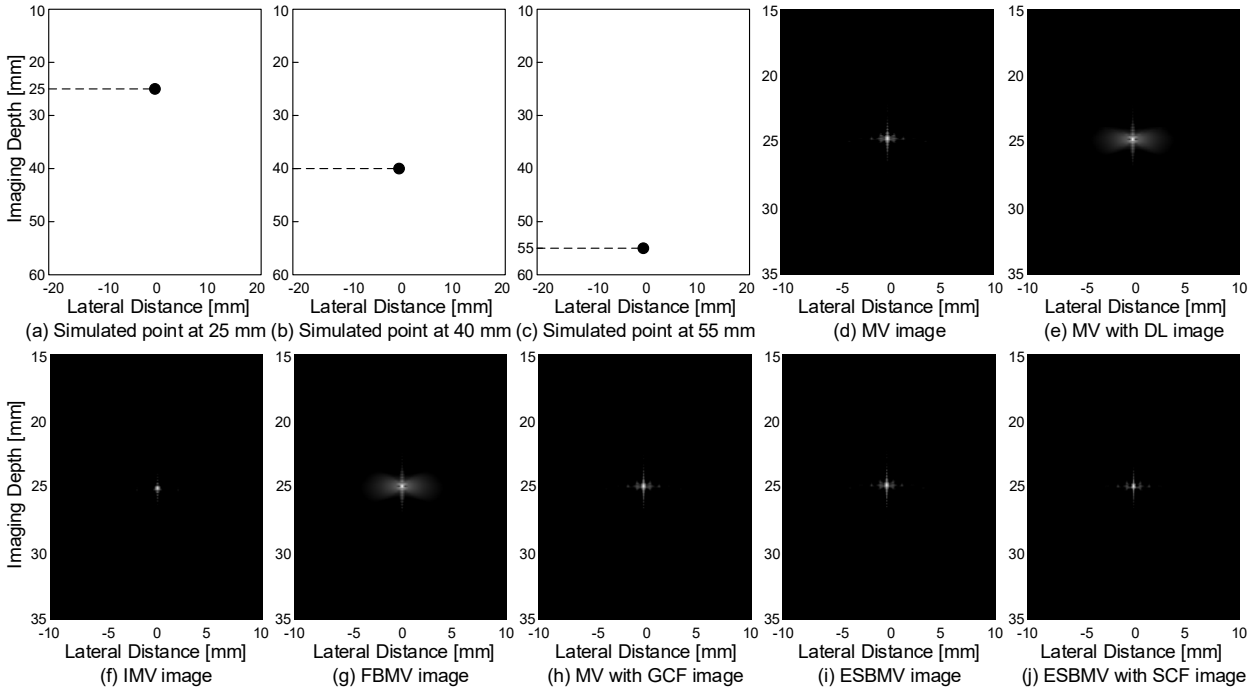

Fig. 10 (a)-(c) Simulated single points at 25 mm, 40 mm and 55 mm. (d)-(j) Output images of various MV adaptive beamforming algorithms for simulated point at 25 mm imaging depth as demonstration examples

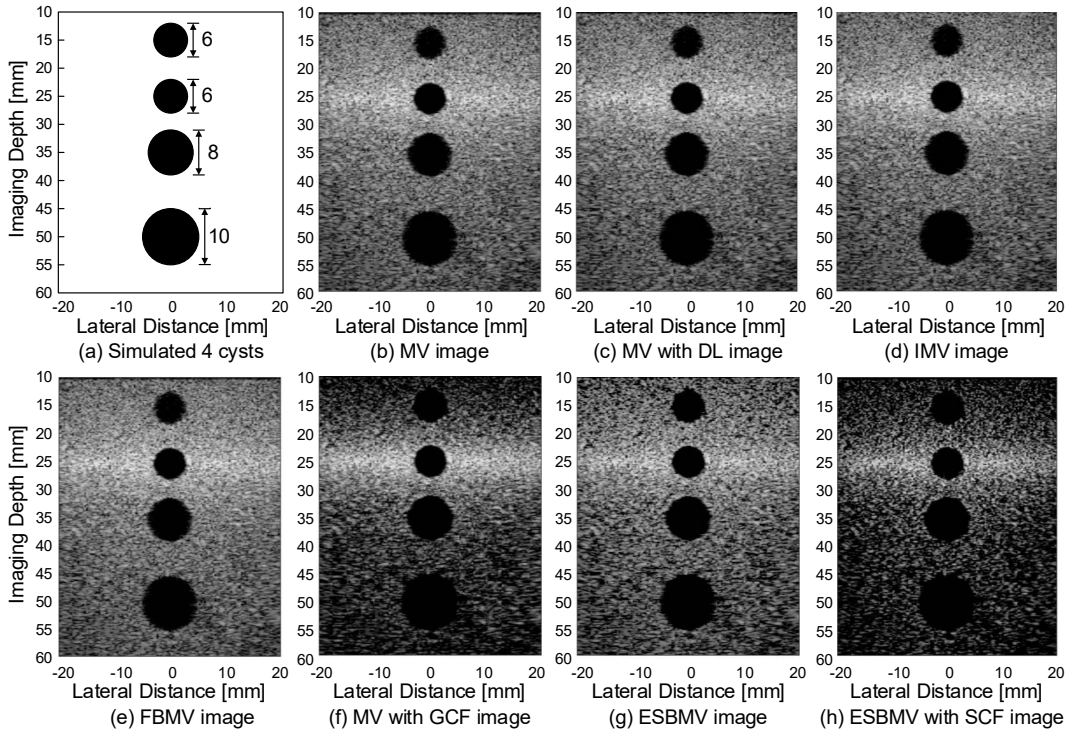

Fig. 11 (a) Simulated cysts at 15 mm, 25 mm, 35 mm and 50 mm with cyst diameters of 6 mm, 6 mm, 8 mm and 10 mm. (b)-(h) Output images of various MV adaptive beamforming algorithms for simulated cysts

trend of the lateral resolution for basic MV algorithm was approaching the better resolution, although there were inconsistencies for some L/M cases, which was because of the relatively poor robustness of basic MV algorithm. But the inconsistencies were within the acceptable lateral resolution value region for algorithm design. Furthermore, the lateral resolution values for various MV beamforming algorithms at 25 mm and 40 mm imaging depth with $M = 64$

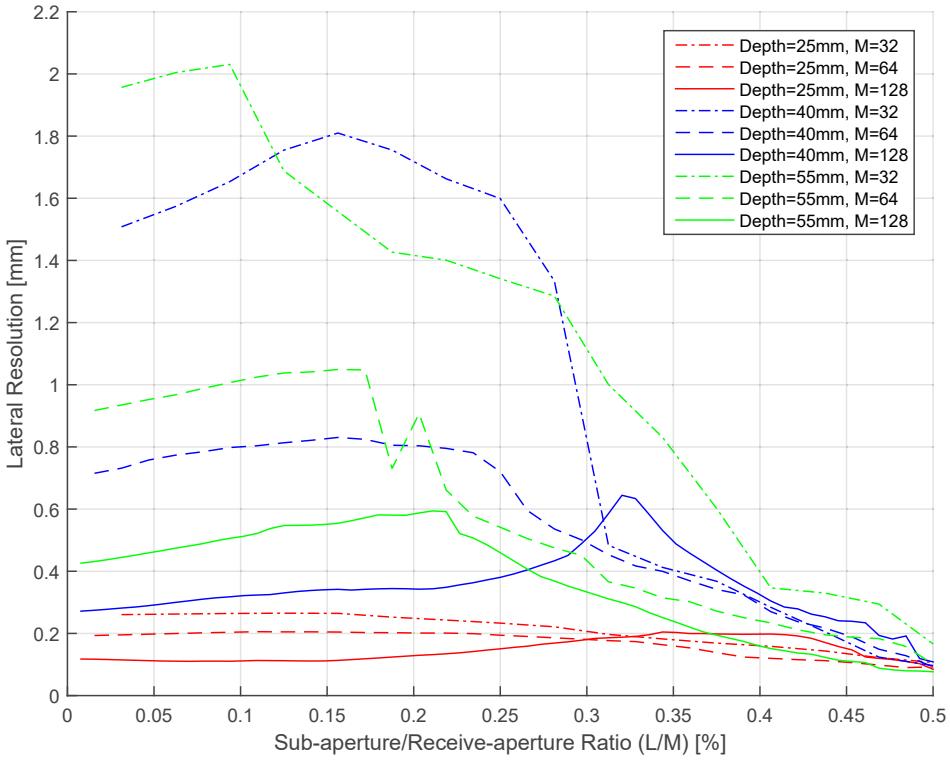

Fig. 12 Lateral resolution values for basic MV beamforming algorithm at 25 mm, 40 mm and 55 mm imaging depths with various M and L

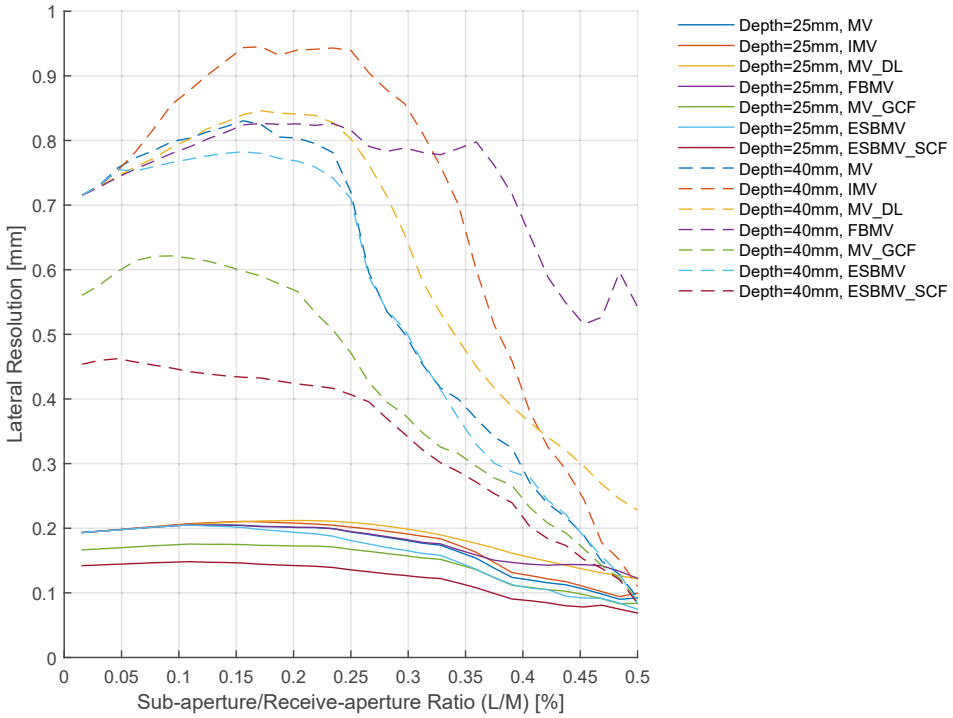

Fig. 13 Lateral resolution values for MV adaptive beamforming algorithms at 25 mm and 40 mm imaging depths with $M = 64$ and various L

are shown in Fig. 13. ESMBV_SCF algorithm presented the best lateral resolution performance among the seven MV algorithms, and MV_GCF algorithm achieved the second best resolution performance. The resolution of basic MV algorithm was generally better than those of IMV, MV_DL and FBMV algorithms and slightly worse than ESBMV algorithm.

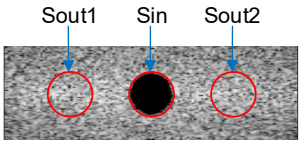

Fig. 14 Illustration of the regions inside and outside the cyst for image contrast evaluation

4.2.2 Quantitative contrast study

The quantitative analysis of image contrast for MV adaptive beamforming algorithms was conducted by comparing the signal amplitude inside and outside the simulated cysts shown in Fig. 11(a). The mathematical expression of the image contrast calculation was: $C = (S_{out} - S_{in})/S_{out}$, where S_{in} was the average signal amplitude inside the area of the cyst, as shown in Fig. 14, and S_{out} was the mean signal amplitude within the same-sized regions outside the cyst. The two same-sized regions outside the cyst were chosen on both sides of the cyst at the same imaging depth (Fig. 14), hence $S_{out} = (S_{out1} + S_{out2})/2$. The image contrast was considered to be better, when the contrast value was larger. The contrast values at various imaging depths for basic MV beamforming algorithm with various M and L are shown in Fig. 15. The better contrast was obtained with larger M when the imaging depth was closer to the focal depth. Besides, while L/M was growing, the contrast of basic MV algorithm was generally degrading with some inconsistencies for some L/M cases. However the inconsistencies were within the acceptable contrast value region for algorithm design. The contrast values for various MV adaptive beamforming algorithms at 25 mm imaging depth with $M = 64$ or $M = 128$ are shown in Fig. 16. ESMBV_SCF and MV_GCF algorithms presented the best contrast performance among the seven MV algorithms at focal depth. ESBMV algorithm demonstrated good performance for a certain range of L/M values but degraded fast after the certain L/M value range. On the other hand, FBMV and MV_DL algorithms had the slowest contrast degradation rates towards larger L/M. Furthermore, basic MV algorithm had the worst contrast performance among the several MV algorithms, and IMV algorithm was the second worst.

4.3 Quantitative study of real-time performance

The real-time performance evaluation of the MV adaptive beamforming algorithms implementation comprised the computational speedup of the GPU implementation over its CPU implementation and the output image frame rate of the algorithm implementation. The evaluation test cases in the experiments included various M and L combinations. The number of the GPU compute blocks were not changed in the experiments, because the number of image pixels, 127×1000 pixels for one image, did not change during the tests. But the number of the GPU compute threads in a block, which was also defined as GPU compute thread block size, was varying during the experiments. In order to find the optimum value of the GPU compute thread block size in the best practices, the computing speed of various GPU compute thread block size was tested. The GPU test compute thread block size case covers all 32 multiplier from 32 to 1024.

The computational speedup of the GPU implementation over its CPU implementation was an important evaluation feature of the MV adaptive beamforming algorithm implementation when use the framework to accelerate the computation process. The overall speedup results of the GPU implementation over the CPU implementation for various M and L combinations were shown in Fig. 17 and Fig. 18. As seen from the figures, in most cases, each algorithm can achieve thousands of times speedup, which fully illustrates the effectiveness of the framework proposed in this paper. In most case, the curves of speedup are very similar, which means that, as a general acceleration framework for MV adaptive beamforming algorithm, it can effectively accelerate the compute process of various MV adaptive beamforming algorithms. Among them, the acceleration of MV_GCF is lower than that of the whole, because the calculation of GCF part contains many complex operations which can not be parallelized, so that the overall acceleration ratio decreases. As L increases, the overall acceleration ratio decreases, because the larger L, the more memory resources it needs, which limits the number of parallel compute blocks and threads, thus reducing the acceleration ratio. Also, as seen from the results, the computational speedup increased when the computational problem size increased because of the larger M value, which means that the acceleration framework with GPU was more efficient when the computational workload was larger.

Apart from the computational speedup, the output image frame rate of beamforming algorithm implementation was calculated as:

$$Frame\ \ rate = \frac{1}{Time_{GPU}} \tag{13}$$

where $Time_{GPU}$ was the algorithm computational time to obtain one frame of image. The output image frame rate results of various algorithms implementation for various M and L combinations were shown in Fig. 19. Horizontal line at the bottom was real-time limit which was defined as 20fps, because we could watch images like a video at this frame rate. As seen from Fig. 19, as the values of M and L increased, the frame rate of the MV adaptive beamforming algorithm increased. because the larger the values of M and L, the longer the algorithm computational time spent because of the increased dramatically computational opera-

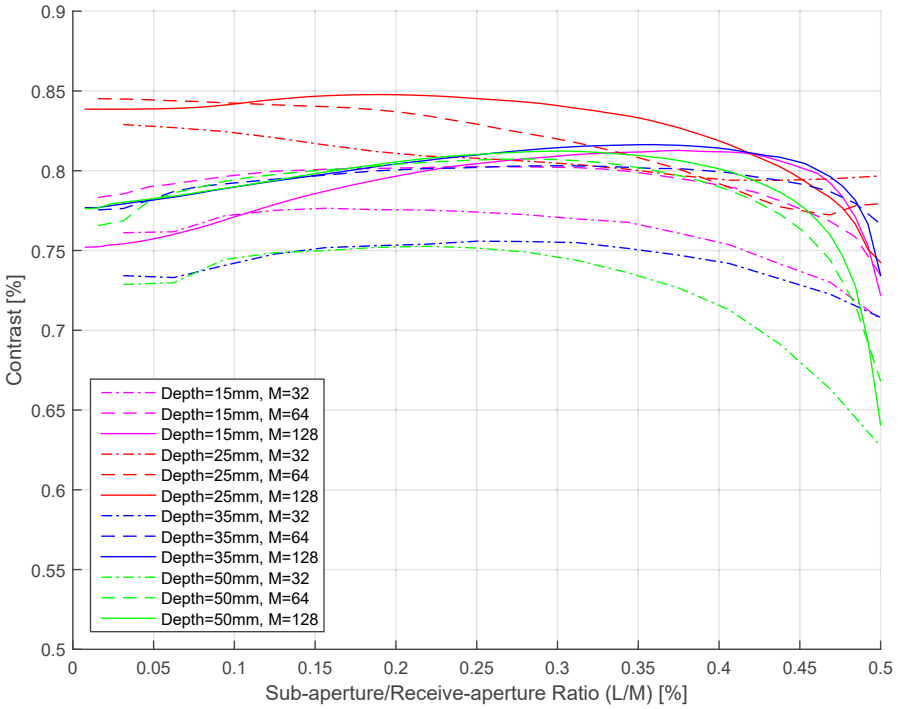

Fig. 15 Contrast values for basic MV beamforming algorithm at 15 mm, 25 mm, 35 mm and 50 mm imaging depths with various M and L

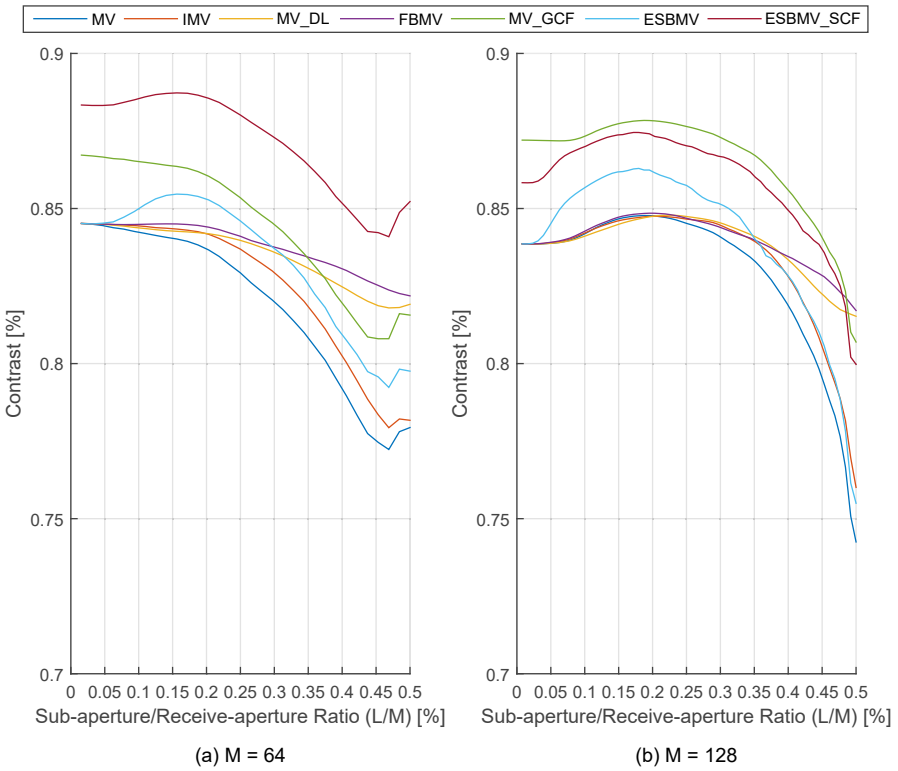

Fig. 16 Contrast values for MV adaptive beamforming algorithms at 25 mm imaging depths with $M = 64$ or 128 and various L

tions. However, in the experiments, many test cases for various M and L combinations achieved real-time video imaging frame rate for medical ultrasonic imaging, which was higher than 25 fps which indicates that the acceleration framework has great significance for realizing the real-time performance of high quality medical ultrasonic imaging.

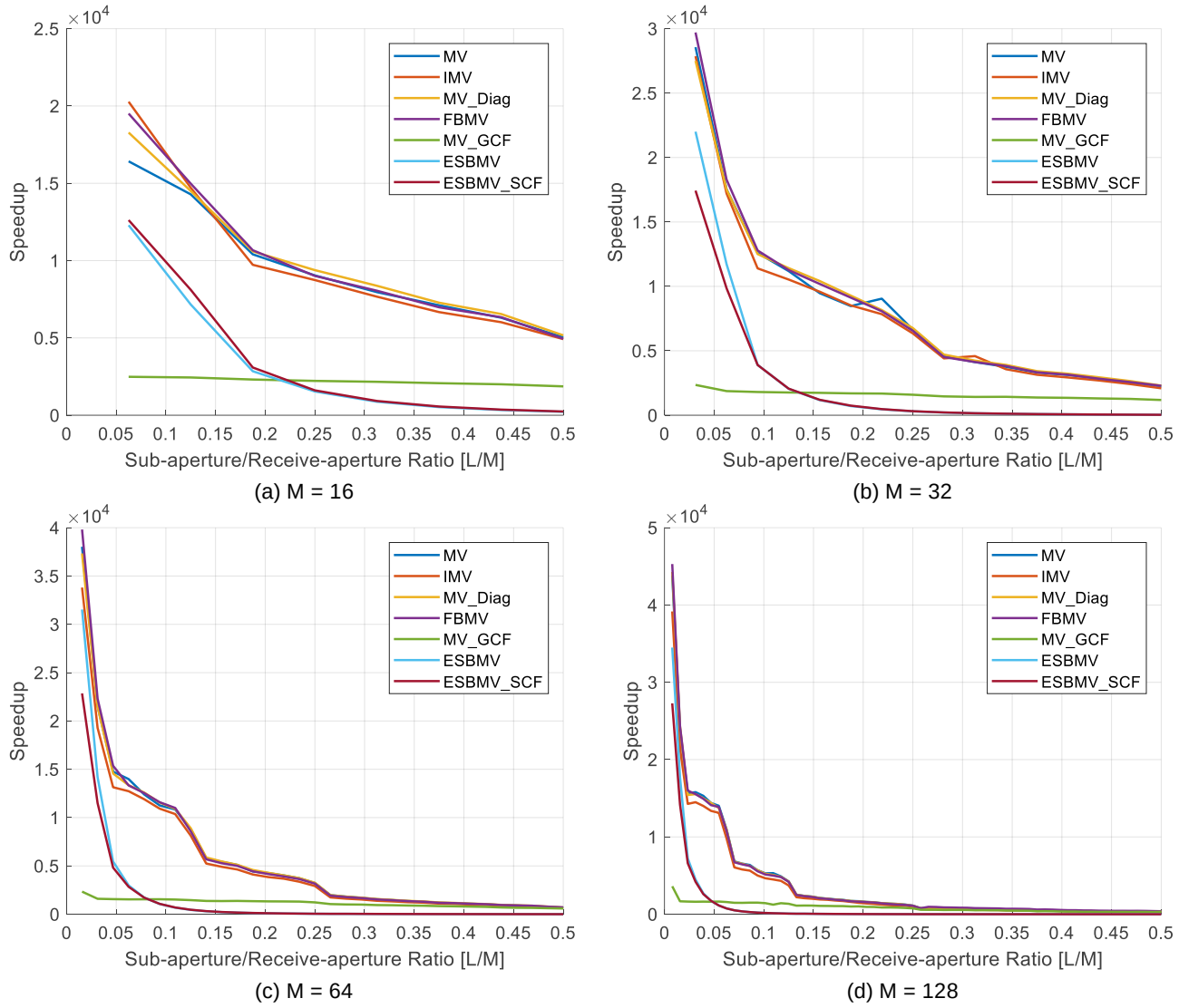

Fig. 17 Speedup values for MV adaptive beamforming algorithms with various M and L

5 Conclusions

An acceleration framework with GPU is proposed to help achieve real-time for minimum variance adaptive beamforming algorithms in medical ultrasonic imaging. The acceleration framework could greatly simplify the parallel implementation of the MV adaptive beamforming algorithms. Various MV adaptive beamforming algorithm implementation was carried out as test cases. The quantitative image quality study of MV adaptive beamforming algorithms for medical ultrasonic imaging was conducted in this work. Based on the study, the image quality performance of various MV adaptive beamforming algorithms was investigated thoroughly. Generally, ESBMV_SCF and MV_GCF algorithms presented better image quality. Although some algorithms (IMV, MV_DL and FBMV) performed worse than basic MV algorithm in lateral resolution analysis, all the algorithms except basic MV algorithm had better image quality than basic MV algorithm. On the other hand, each algorithm implemented with the acceleration could achieve thousands of times speedup and could fulfill the real-time imaging requirements for medical ultrasonic imaging in many test cases. Moreover, The image quality trends of lateral resolution and contrast for MV adaptive beamforming algorithms can provide good guidance for proper trade-off designs of real-time MV beamforming algorithm implementations, by setting specific image quality acceptable region based on the quantitative study. On the other hand, as MV adaptive beamforming is gradually applied to other fields, like photoacoustic imaging, this framework may be applied to these fields for acceleration.

Acknowledgements This work is supported by "Special Program for Applied Research on Super Computation of the NSFC-Guangdong Joint Fund (the second phase), "Guangdong Natural Science Foundation"

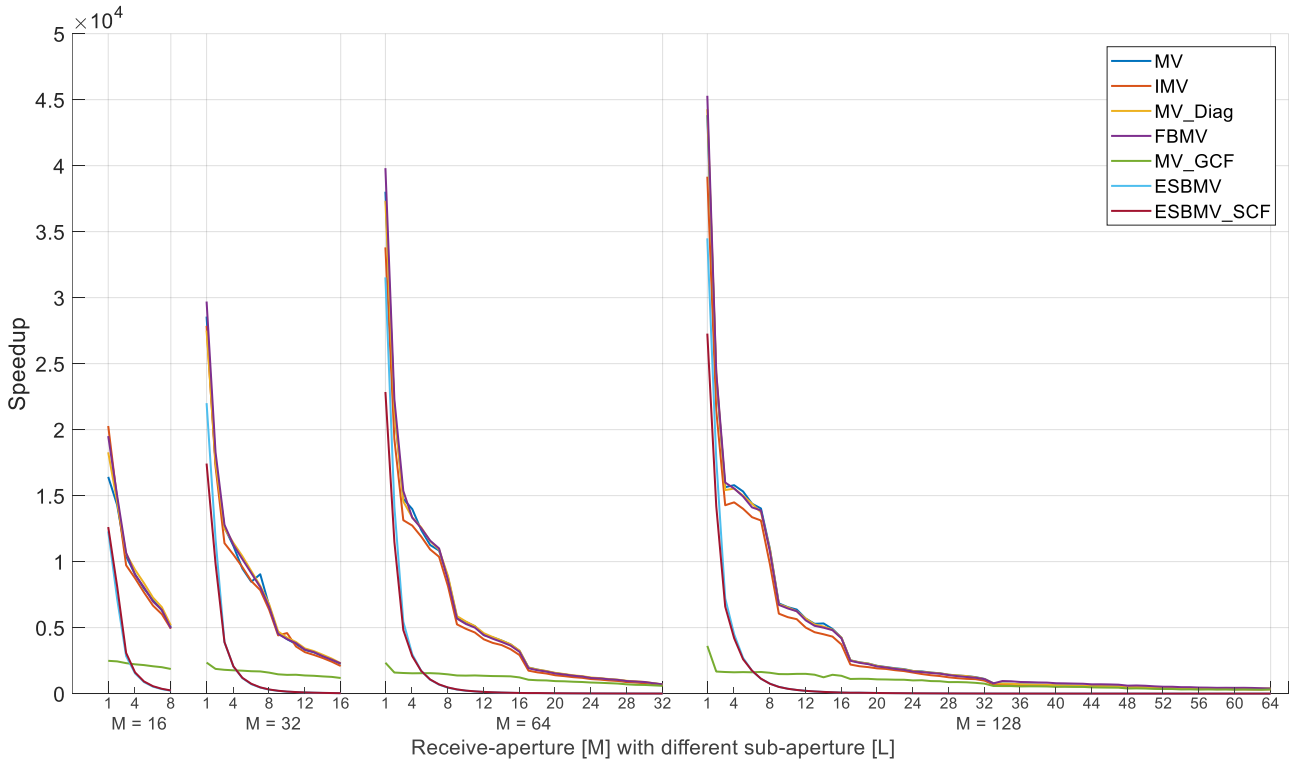

Fig. 18 Speedup values for MV adaptive beamforming algorithms with various M and L

(No. 2016A030310412), "Guangzhou Science and Technology Program" (Key Laboratory Project No. 15180007) and "the Fundamental Research Funds for the Central Universities" (No. 2015ZM081).

References

1. Asl, B.M., Mahloojifar, A.: Eigenspace-based minimum variance beamforming applied to medical ultrasound imaging. IEEE Transactions on Ultrasonics Ferroelectrics & Frequency Control **57**(11), 2381–2390 (2010)
2. Asl, B.M., Mahloojifar, A.: Contrast enhancement and robustness improvement of adaptive ultrasound imaging using forward-backward minimum variance beamforming. IEEE Transactions on Ultrasonics Ferroelectrics & Frequency Control **58**(4), 858–867 (2011)
3. Babak Mohammadzadeh, A., Ali, M.: Minimum variance beamforming combined with adaptive coherence weighting applied to medical ultrasound imaging. IEEE Transactions on Ultrasonics Ferroelectrics & Frequency Control **56**(9), 1923–1931 (2009)
4. Capon, J.: High-resolution frequency-wavenumber spectrum analysis. Proceedings of the IEEE **57**(8), 1408–1418 (1969)
5. Chen, J., Chen, J., Min, H.: Design and evaluation of medical ultrasonic adaptive beamforming algorithm implementation on heterogeneous embedded computing platform. Eurasip Journal on Embedded Systems **2017**(1), 19 (2017)
6. Deylami, A.M., Jensen, J.A., Asl, B.M.: An improved minimum variance beamforming applied to plane-wave imaging in medical ultrasound. In: Ultrasonics Symposium, pp. 1–4 (2016)
7. Havlice, J.F., Taenzer, J.C.: Medical ultrasonic imaging: An overview of principles and instrumentation. Proceedings of the IEEE **67**(4), 620–641 (1979)
8. Jensen, J.A., Svendsen, N.B.: Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers. IEEE Transactions on Ultrasonics Ferroelectrics & Frequency Control **39**(2), 262–267 (1992)
9. Li, H., Li, J., Stoica, P.: Performance analysis of forward-backward matched-filterbank spectral estimators. IEEE Transactions on Signal Processing **46**(7), 1954–1966 (1998)
10. Li, J., Stoica, P., Wang, Z.: On robust capon beamforming and diagonal loading. IEEE Transactions on Signal Processing **51**(7), 1702–1715 (2003)
11. Li, P.C., Li, M.L.: Adaptive imaging using the generalized coherence factor. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control **50**(2), 128–141 (2003)
12. Liu, T., Zhao, H., Zheng, Y.: Eigenspace-based minimum variance beamforming combined with sign coher-

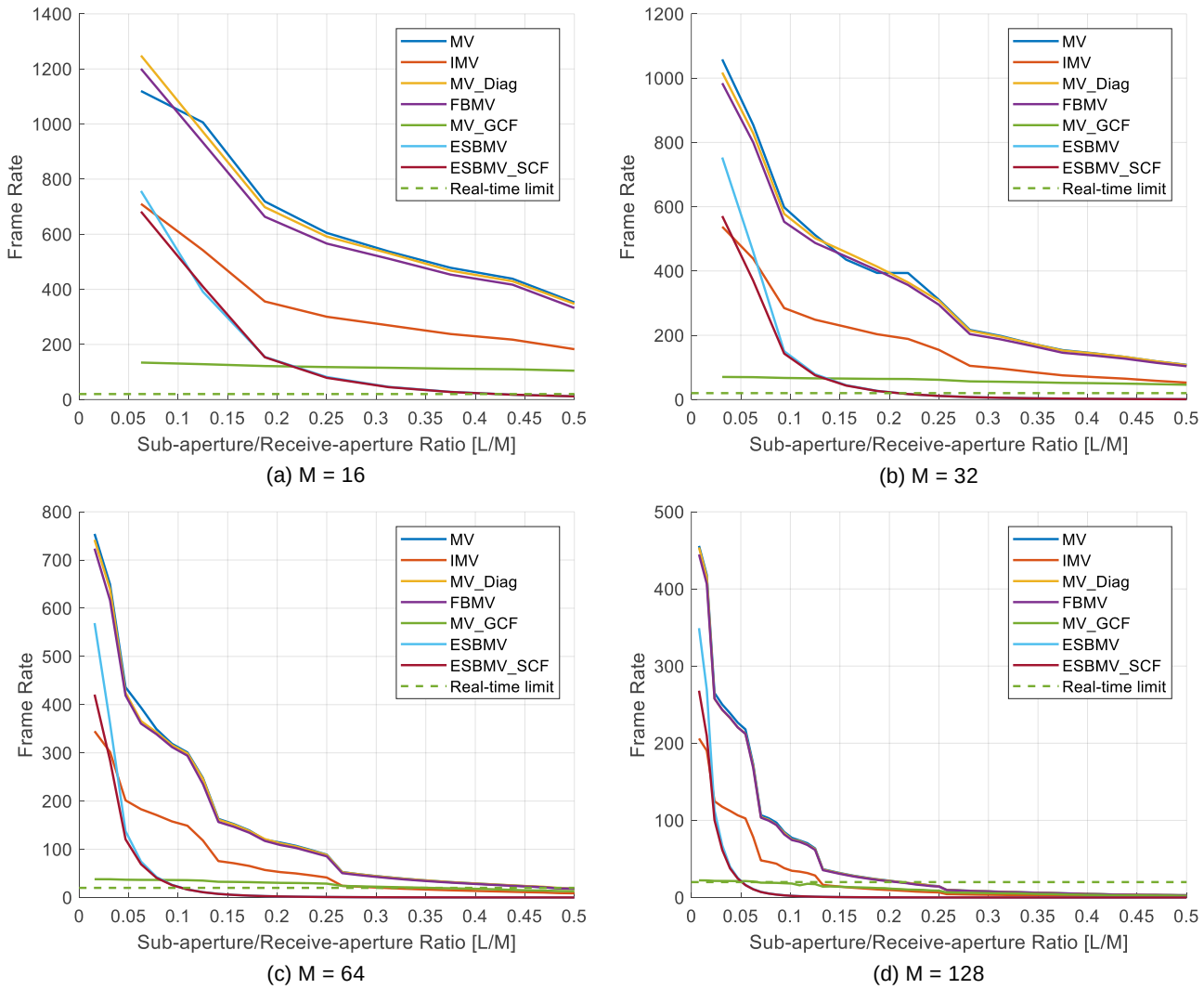

Fig. 19 Frame rate values for MV adaptive beamforming algorithms with various M and L

ence factor for ultrasound beamforming (in chinese). Acta Acustica **40**(6), 855–862 (2015)
13. Matrone, G., Ramalli, A., Savoia, A.S., Tortoli, P., Magenes, G.: High frame-rate, high resolution ultrasound imaging with multi-line transmission and filtered-delay multiply and sum beamforming. IEEE Transactions on Medical Imaging **36**(2), 478–486 (2017)
14. Mozaffarzadeh, M., Mahloojifar, A., Orooji, M., Kratkiewicz, K., Adabi, S., Nasiriavanaki, M.: Linear-array photoacoustic imaging using minimum variance-based delay multiply and sum adaptive beamforming algorithm. Journal of Biomedical Optics **23**(2), 1–15 (2018)
15. Mozaffarzadeh, M., Mahloojifar, A., Periyasamy, V., Pramanik, M., Orooji, M.: Eigenspace-based minimum variance combined with delay multiply and sum beamformer: Application to linear-array photoacoustic imaging. IEEE Journal of Selected Topics in Quantum Electronics **25**(1), 1–8 (2018)
16. Nai, S.E., Ser, W., Yu, Z.L., Chen, H.: Iterative robust minimum variance beamforming. IEEE Transactions on Signal Processing **59**(4), 1601–1611 (2011)
17. Qi, Y., Wang, Y., Guo, W.: Joint subarray coherence and minimum variance beamformer for multitransmission ultrasound imaging modalitie. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control **65**(9), 1600–1617 (2018)
18. Qi, Y., Wang, Y., Yu, J., Guo, Y.: 2-d minimum variance based plane wave compounding with generalized coherence factor in ultrafast ultrasound imaging. Sensors **18**(12), 4099 (2018)
19. Qiu, L., Cai, Y., Zhao, M.: Low-complexity variable forgetting factor mechanisms for adaptive linearly constrained minimum variance beamforming algorithms. Signal Processing Iet **9**(2), 154–165 (2015)

20. Synnevag, J., Austeng, A., Holm, S.: Adaptive beamforming applied to medical ultrasound imaging. IEEE Transactions on Ultrasonics Ferroelectrics & Frequency Control **54**(8), 1606–1613 (2007)
21. Synnevag, J., Austeng, A., Holm, S.: Benefits of minimum-variance beamforming in medical ultrasound imaging. IEEE Transactions on Ultrasonics Ferroelectrics & Frequency Control **56**(9), 1868–1879 (2009)
22. Wu, W., Pu, J., Lv, Y.: Medical ultrasound imaging method combining minimum variance beamforming and general coherence factor. Acta Acustica **36**(1), 66–72 (2011)
23. Zhao, J., Wang, Y., Yu, J., Guo, W., Li, T., Zheng, Y.P.: Subarray coherence based postfilter for eigenspace based minimum variance beamformer in ultrasound plane-wave imaging. Ultrasonics **65**, 23–33 (2016)
24. Zheng, C., Peng, H.: Ultrasound imaging algorithm by combining eigenspace-based minimum variance beamforming and correlation coeficient eigenvalue weighting. Acta Acustica **46**(1), 25–29 (2016)
25. Ziksari, M.S., Asl, B.M.: Combined phase screen aberration correction and minimum variance beamforming in medical ultrasound. Ultrasonics **75**, 71–79 (2017)