

SEPARABLE BEAMFORMING FOR 3-D SYNTHETIC APERTURE ULTRASOUND IMAGING

Ming Yang, Richard Sampson*, Thomas F. Wenisch* and Chaitali Chakrabarti

SECEE, Arizona State University, Tempe, AZ 85287

*EECS, University of Michigan, Ann Arbor, MI 48109

ABSTRACT

Recently, there has been great interest in 3-D ultrasound imaging, but power constraints have precluded practical implementation of high-resolution 3-D ultrasound in handheld imaging platforms. In this paper, we propose a separable beamforming procedure for subaperture-based 3-D synthetic aperture ultrasound systems that drastically reduces computational (and hence power) requirements. Separable beamforming approximates 2-D array beamforming through a series of 1-D beamforming operations. Our proposed method is based on a separable delay decomposition method that minimizes phase error. We show that the proposed separable system achieves 19-fold complexity reduction while producing images with contrast-to-noise ratio comparable to that of non-separable systems. Furthermore, we propose an online iterative delay calculation method that substantially reduces storage requirements. We briefly describe how the separable beamformer can be implemented in the context of Sonic Millip3De, our recently proposed hardware accelerator for the digital front-end of a 3-D ultrasound system.

Index Terms— Separable beamforming, separable delay, 2-D array, hardware implementation

1. INTRODUCTION

Ultrasound imaging is one of the most popular medical imaging modalities; it is inexpensive relative to CT and MRI and poses no known side-effects. Unlike other imaging modalities, generating and sensing ultrasound signals does not require high power, which makes handheld ultrasound imaging systems feasible.

Recently, there has been a great deal of interest in development of 3-D ultrasound imaging systems which generate volumetric images that are easier to interpret and lead to better diagnosis [1]. However, implementing a high-resolution and high-image-quality 3-D ultrasound system within the power budget constraints of a handheld device is challenging, and no commercial offerings of hand-held 3-D ultrasound yet exist. We recently proposed Sonic Millip3De [2], a 3-D die-stacking hardware accelerator for ultrasound beamforming, which is able to generate a 3-D volume with 45° in both azimuth and elevation view angles and 10cm maximum depth at 1 frame/second within a 16W system power budget in 45nm technology. Sonic Millip3De leverages subaperture processing [3, 4] and a narrow-bit-width streaming hardware architecture that eliminates much of the computational complexity of delay calculation. More recently, we proposed an optimized subaperture apodization and firing scheme which reduces the number of firings by half and hence the system power consumption [5]. Nevertheless, the existing design still falls short of desirable power targets (e.g., 5W for safe contact with human skin), calling for further reductions in front-end processing requirements.

Processing might be further reduced by utilizing sparse 2-D arrays [6, 7], but at significant cost in image quality due to artifacts from grating lobes and larger sidelobes.

An alternative approach to reduce front-end complexity is to use separable beamforming [8, 9], wherein conventional 2-D array beamforming is decomposed into a series of 1-D beamforming problems at the cost of some reduction in image quality. A frequency-domain separable beamforming method was proposed in [8] that is amenable to parallel execution but can not support dynamic focusing, and has lower image quality. An X - Y separable beamforming for systems with rectangular scan geometry (scanlines are perpendicular to the 2-D array plane) was proposed in [9]. However this method is applicable only to rectangular scan systems, which generate views that are too narrow for large anatomy. Moreover, the proposed method relies on a firing scheme that limits lateral resolution due to the small subaperture size.

In this paper, we propose a new separable beamforming method that is not restricted to any specific scan system. It is based on a delay decomposition method that minimizes the root-mean-square (RMS) phase error introduced by the separable delay approximation. We show how this method can be used to achieve separable beamforming in subaperture based 3-D synthetic aperture ultrasound (SAU) systems with 19-fold reduction in computation complexity compared to the non-separable method. The separable delays obtained by this method could be stored in large look-up tables and directly used for delay calculations. However, to reduce look-up table storage requirements, we propose a method to iteratively compute these delays at runtime using far fewer pre-computed constants. We demonstrate, using Field II simulations of cyst images, that our proposed separable beamforming method achieves almost identical image quality compared to the non-separable baseline method.

We incorporate the separable beamforming algorithm into the existing Sonic Millip3De design. The $19\times$ reduction in computation can be leveraged either to reduce hardware requirements (by scaling down the number of parallel pipelines in the design), increase frame rate (by operating the existing design at higher throughput), or save power (by scaling down frequency to maintain current frame rates). We briefly describe the architectural modifications that are required to support 2-stage separable beamforming on Sonic Millip3De. These modifications include additional SRAM storage and reorganization of the 1,024 pipelines into 32 clusters with 32 nodes per cluster to facilitate the 2-stage beamforming.

The remainder of this paper is organized as follows. In Section 2, we introduce the proposed beamforming delay decomposition method which minimizes RMS phase error followed by the iterative delay calculation method. We report on our image quality analysis in Section 3. Next, in Section 4, we describe the modifications to Sonic Millip3De required to support the separable beamformer. Finally, we conclude in Section 5.

This work was funded in part by grant NSF-CSR0910699.

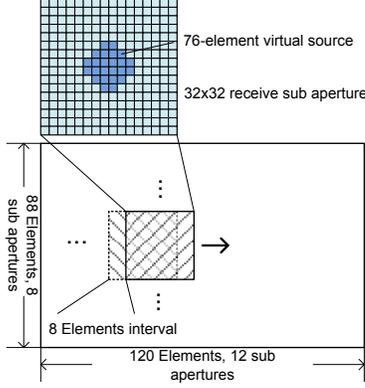


Fig. 1: 2-D subaperture based SAU firing and receive

2. ALGORITHM

Although 3-D ultrasound systems can provide high-quality 3-D images, making diagnoses faster and more accurate, the improved imaging capability comes at a high cost in computational complexity. 3-D systems must process far more scanlines and focal points than their 2-D counterparts and process input signals from many more transducers. The system parameters of the 3-D ultrasound system studied in this paper are shown in Table 1. The transducer array of size 120×88 contains over 10 thousand transducers and could generate over 400 million samples per second assuming 40MHz A/D sampling rate. To make processing requirements tractable, existing 3-D systems must compromise on resolution, image quality, or frame rate.

Table 1: System parameters

Property	Value
Pitch, μm	192.5
Array size, element	120×88
Subaperture size, element	32×32
Number of scanlines	48×48
View angle, square degree	$45^\circ \times 45^\circ$
Max depth, cm	10
Center frequency, MHz	4
6dB transducer bandwidth, MHz	4
A/D sampling rate, MHz	40

2.1. Subaperture Processing

Subaperture-based processing is one way of reducing the number of concurrent active channels, thereby reducing the computation load per firing. The subaperture firing scheme used in this paper is illustrated in Fig. 1. 120×88 array elements are organized into 96 overlapping subapertures of 32×32 elements each. All 32×32 elements are used for receive, while for transmit, only 76 array elements in the center of the subaperture are activated during firing. This scheme simulates a single virtual source located above the array plane. The 96 subapertures (8 rows and 12 columns) fire and receive in turn to cover all 120×88 array elements; the center of adjacent sub-apertures are 8 elements apart. The 3-D images generated at the end of each fire and receive sequence are summed to produce the final image.

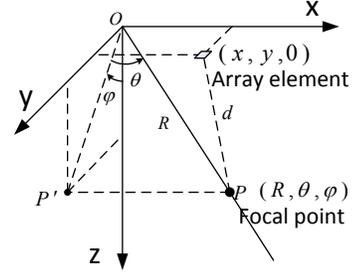


Fig. 2: Array and beamforming coordinate system

2.2. Separable Beamforming Process

The 3-D coordinate system used in our scheme is shown in Fig. 2. Let (R, θ, ϕ) be the coordinates of a focal point P on a scanline. Here R is the radial distance from the origin O to point P . Point P' is the orthogonal projection of point P in the yz plane. ϕ is the elevation angle between line OP' and z axis. θ is the azimuth angle between OP and its orthogonal projection OP' in the yz plane. For a transducer array element at $(x, y, 0)$, the distance between the transducer element and the focal point P is given by

$$d = \sqrt{R^2 + x^2 - 2Rx \sin(\theta) + y^2 + 2Ry \cos(\theta) \sin(\phi)} \quad (1)$$

Assuming that the ultrasound speed is c , and the round-trip delay between the origin and the focal point is $2R/c$, the round-trip delay at the transducer relative to that at the origin is given by $\tau(x, y, R, \theta, \phi) = (2R - d_{\text{transmit}} - d_{\text{receive}})/c$. Let $\tau(n_x, n_y, m_R, m_\theta, m_\phi)$ be the discrete form of $\tau(x, y, R, \theta, \phi)$, where n_x and n_y are variables associated with the coordinates of receive elements, and m_R , m_θ and m_ϕ are variables associated with the coordinates of focal points. Then the non-separable beamforming corresponding to subaperture l is described as

$$F_l(m_R, m_\theta, m_\phi; t) = \sum_{n_x=1}^{N_x} \sum_{n_y=1}^{N_y} A_l(n_x, n_y) S_l(n_x, n_y, t - \tau(n_x, n_y, m_R, m_\theta, m_\phi)) \quad (2)$$

where $S_l(n_x, n_y, t)$ is the signal received by transducer element (n_x, n_y) in subaperture l , and $A_l(n_x, n_y)$ is the corresponding apodization coefficient. $F_l(m_R, m_\theta, m_\phi; t)$ is the low resolution 3-D image generated by subaperture l . F_l should be sampled at $t = 2R/c$ for dynamic focusing.

Now, if $\tau(n_x, n_y, m_R, m_\theta, m_\phi)$ can be represented by the sum of $\tau_1(n_x, n_y, m_R, m_\theta)$ and $\tau_2(n_y, m_R, m_\theta, m_\phi)$, equation (2) can be represented by a two-stage separable beamforming process.

$$F_l^{(1)}(n_y, m_R, m_\theta; t) = \sum_{n_x=1}^{N_x} A_l(n_x, n_y) S_l(n_x, n_y, t - \tau_1(n_x, n_y, m_R, m_\theta))$$

$$F_l^{(2)}(m_R, m_\theta, m_\phi; t) = \sum_{n_y=1}^{N_y} F_l^{(1)}(m_R, n_y, m_\theta; t - \tau_2(n_y, m_R, m_\theta, m_\phi)) \quad (3)$$

Here $F_l^{(1)}$ is the partially beamformed intermediate signal

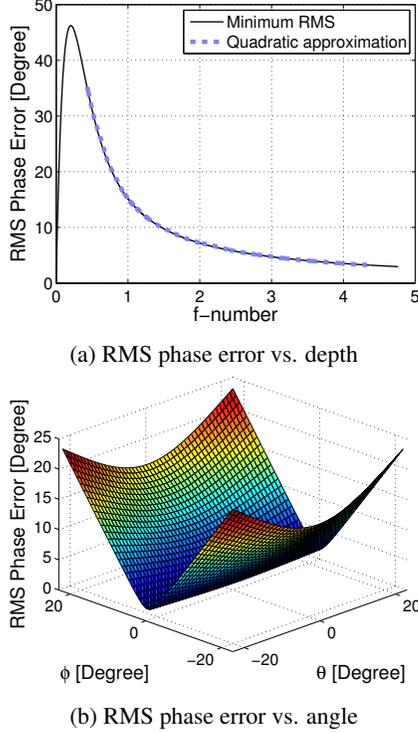


Fig. 3: RMS error of separable delay based method

obtained by 1-D beamforming along the x axis for all combinations of m_R , n_y and m_θ . In the second stage, 1-D beamforming is performed along the y axis for all combinations of m_R , m_θ and m_ϕ . Note that, similar to non-separable beamforming, separable beamforming results should be sampled at $t = 2R/c$.

The number of delay-sum operations of separable beamforming for one subaperture is $N_x N_y M_R M_\theta + N_y M_R M_\theta M_\phi$ in contrast to $N_x N_y M_R M_\theta M_\phi$ in non-separable beamforming. Thus the computation complexity reduction is $N_x M_\phi / (N_x + M_\phi)$. For the configuration shown in Table 1 with 32×32 subaperture and 48×48 scanlines, our system achieves about $19\times$ complexity reduction.

The separable beamforming method is based on the assumption that the wave propagation delay τ can be decomposed into τ_1 and τ_2 . However the decomposition is not exact and the performance of this method depends on the accuracy of this approximation. Next we describe the proposed decomposition method based on minimizing RMS phase error.

2.3. Delay Approximation and Calculation

2.3.1. Separable Delay with Minimum RMS Error

The beamsum delay τ , which is a function of five variables, cannot be strictly decomposed into a sum of two functions with fewer variables because distance calculation involves the square root operation (as shown in equation (1)). The Taylor series of the square root function comprises functions involving multiple variables, also referred to as cross terms, which can not be easily decomposed. To make the delay equation separable, some of these cross terms must be dropped. Although the effect of the cross terms diminish with large R , for small f -numbers, the cross terms in the delay calculation cannot be neglected.

Consider a simple delay decomposition that is given by

$\tau(n_x, n_y, m_R, m_\theta, m_\phi) = \tau_1(n_x, m_R, m_\theta) + \tau_2(n_y, m_R, m_\phi)$, where both τ_1 and τ_2 are functions of three variables. The justification for this decomposition is as follows. For dynamic focusing, both τ_1 and τ_2 depend on m_R . τ_1 is also a function of m_θ and n_x because 1-D beamforming along x direction allows the array system to distinguish signals coming from different azimuth angles. Hence m_θ and n_x should not be separated. Similarly, 1-D beamforming along y direction allows the array system to distinguish signals coming from different elevation angles, so m_ϕ and n_y should not be separated. Unfortunately this simple decomposition has large errors primarily because θ and ϕ are separated and the cross term involving θ and ϕ is lost.

One way to solve this problem is by increasing the number of variables of τ_1 and τ_2 . For τ_1 , consider adding m_ϕ or n_y to the variable list already consisting of m_R , n_x and m_θ . There is no benefit in adding m_ϕ because 1-D beamforming in the first stage is along the x direction, and does not have enough resolution along ϕ . However there is an advantage of adding n_y to the variable list of τ_1 . This is because τ_1 already includes n_x and thus adding n_y helps to retain cross terms between n_x and n_y . For τ_2 , there are two candidate variables that can be added to the variable list, namely n_x and m_ϕ . Adding n_x to τ_2 is not a good option because neither the input signal of second-stage beamforming $F^{(1)}$ nor output of second stage beamforming $F^{(2)}$ contains n_x . However adding m_θ to τ_2 has the benefit of preserving cross terms involving θ and ϕ . By increasing the number of variables of τ_1 and τ_2 from three variables to four variables, the approximation error is reduced by one decade. Moreover, equation (3) shows that the four variable version of the decomposition does not increase the computational complexity of separable beamforming.

The next question is how to generate $\tau_1(n_x, n_y, m_R, m_\theta)$ and $\tau_2(n_y, m_R, m_\theta, m_\phi)$ such that the error due to approximation is minimized. Let the RMS error E be represented by

$$E = \sum_{n_x=1}^{N_x} \sum_{n_y=1}^{N_y} \sum_{m_R=1}^{M_R} \sum_{m_\theta=1}^{M_\theta} \sum_{m_\phi=1}^{M_\phi} [\tau(n_x, n_y, m_R, m_\theta, m_\phi) - (\tau_1(n_x, n_y, m_R, m_\theta) + \tau_2(n_y, m_R, m_\theta, m_\phi))]^2 \quad (4)$$

We take the partial derivatives with respect to $\tau_1(n_x, n_y, m_R, m_\theta)$ and $\tau_2(n_y, m_R, m_\theta, m_\phi)$ for all n_x , n_y , m_R , m_θ and m_ϕ and set the partial derivatives to zero. We find that the optimization for the total RMS error is equivalent to individually minimizing the RMS error for each combination of (n_y, m_R, m_θ) . By solving a set of $N_x M_\phi$ linear equations for each combination of (n_y, m_R, m_θ) , we find that the minimum RMS error is achieved when τ_1 and τ_2 are generated according to equations (5) and (6).

$$\tau_1(n_x, n_y, m_R, m_\theta) = \frac{1}{M_\phi} \sum_{m_\phi=1}^{M_\phi} \tau(n_x, n_y, m_R, m_\theta, m_\phi) - \rho(n_y, m_R, m_\theta) \quad (5)$$

$$\tau_2(n_y, m_R, m_\theta, m_\phi) = \frac{1}{N_x} \sum_{n_x=1}^{N_x} \tau(n_x, n_y, m_R, m_\theta, m_\phi) - \rho(n_y, m_R, m_\theta) \quad (6)$$

$$\rho(n_y, m_R, m_\theta) = \frac{1}{N_x M_\phi} \sum_{n_x=1}^{N_x} \sum_{m_\phi=1}^{M_\phi} \tau(n_x, n_y, m_R, m_\theta, m_\phi) \quad (7)$$

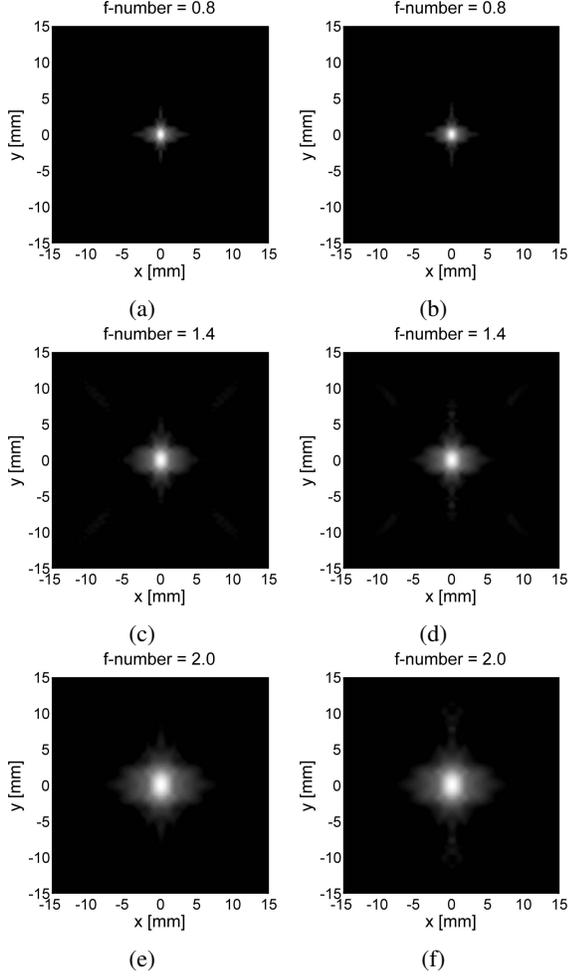


Fig. 4: Simulated xy -plane projection of point-spread function in different depths. Subplots (a), (c) and (e) correspond to non-separable beamforming, and subplots (b), (d) and (f) correspond to separable beamforming. Display dynamic range is 65dB.

Note that equations (5) and (6) are not the only forms that could achieve minimum RMS error. We choose $\rho(n_y, m_R, m_\theta)$ to be of this form because it helps the average value of τ_1 and τ_2 to be similar. From an architectural perspective, this makes the delay line length or buffer depth roughly equal in the two beamforming stages.

2.3.2. Online Separable Delay Calculation

Next we focus on efficient calculation of τ_1 and τ_2 . It is not practical to generate τ_1 and τ_2 using equations (5) and (6) in real time because it requires calculations for all possible values for $\tau(n_x, n_y, m_R, m_\theta, m_\phi)$. While look-up tables for τ_1 and τ_2 can be generated off-line using MATLAB, storing τ_1 and τ_2 as look-up tables is also not practical. For our system configuration, the direct look-up tables for τ_1 and τ_2 contain at least (considering symmetry) 5.7 billion and 8.9 billion constants, respectively.

Fortunately, the delay values of consecutive samples on a scanline do not change much. Hence it is possible to iteratively calculate the delay value for the i th focal point on a scanline from the delay value of the $(i - 1)$ th focal point. We propose to use piece-wise quadratic curves to approximate delay difference

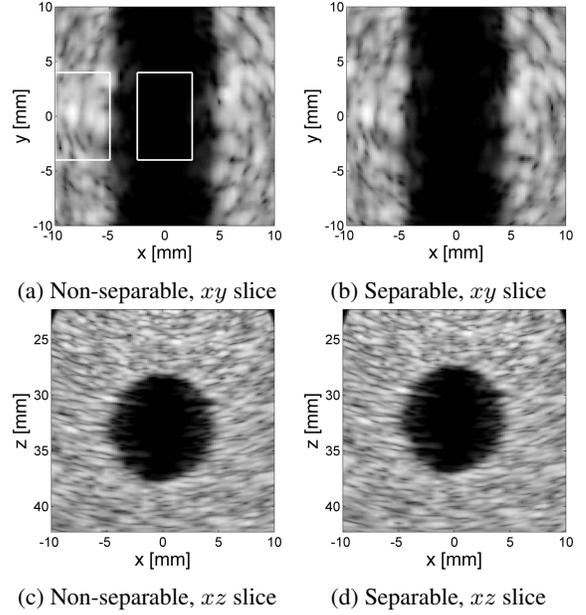


Fig. 5: 10-mm-diameter anechoic cylinder slices with 50dB dynamic range

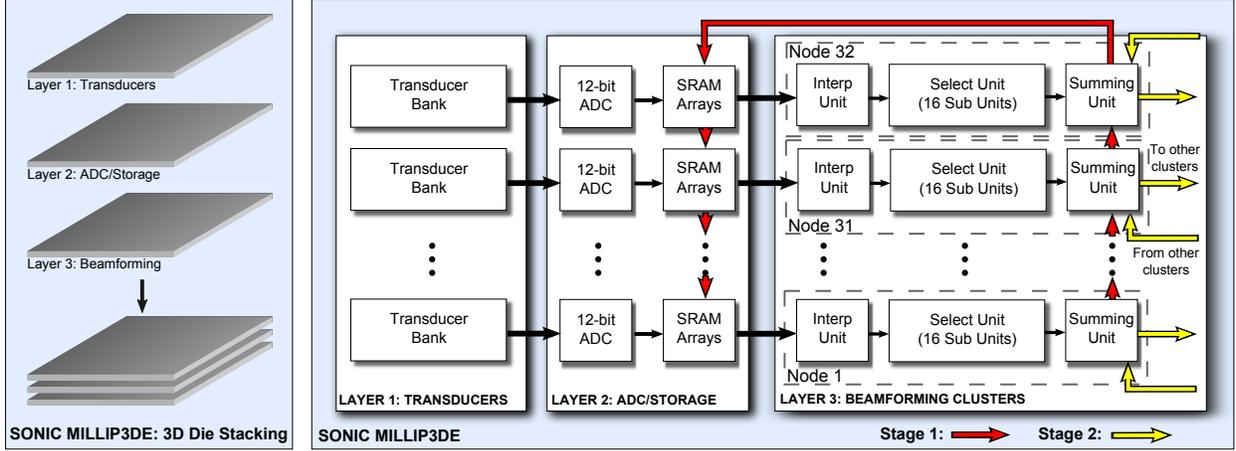
between consecutive samples along a scanline. For example, let $\tilde{\tau}(m_R)$ be the delay corresponding to the m_R th focal point for fixed n_x and n_y . Let $\eta(m_R) = \tilde{\tau}(m_R + 1) - \tilde{\tau}(m_R)$, then $\eta(m_R)$ can be approximated by $am_R^2 + bm_R + c$. Instead of storing the look-up table directly, the coefficients a , b and c and the initial delay are stored, and the delays are calculated from those coefficients. The iterative calculation method does not need multiplications, but requires three additions and a simple circuit for implementation. A similar method has been used in [2] for non-separable beamforming.

To get an accurate approximation, each scanline is divided in to 2-4 sections and the delays in each scanline can be approximated by quadratic curves for each section. For our system configuration, where the depth ranges from 2cm to 10cm, we choose 3 sections, each characterized by three constants and an initial point. 34M constants must be stored; 14M constants are required for τ_1 and the remaining 20M for τ_2 . The 14M constants for τ_1 correspond to 4 constants/section \times 3 sections/scanline \times 48 scanlines \times 1,024 transducers/subaperture \times 96 subapertures, divided by 4 due to symmetry. The number of constants for τ_2 is calculated in a similar way. Each constant requires 12 bits [2], resulting an overall storage requirement of 51MB.

3. SIMULATION RESULTS

To evaluate the delay errors due to the separable delay approximation, we compared delays generated by the proposed method with that of the non-separable delay method. We used Field II [10, 11] to run simulations in MATLAB. Our system configuration is listed in Table 1.

Fig. 3a shows the RMS phase error as a function of f-number. We assume a fixed full aperture size of 120×88 , hence f-number is proportional to depth. The thin solid line in Fig. 3a is the minimum RMS phase error corresponding to separable delays generated according to equations (5) and (6). The largest phase error is in the near field where f-number is about 0.3. As the f-number increases, the phase error decreases rapidly. The phase error drops below 18°



(a) 3 layer die stacking

(b) Pipeline of a single 32 node cluster

Fig. 6: Design of Sonic Millip3De for separable beamforming. Layer 3 contains 32 clusters with 32 nodes per cluster.

when f-number is larger than 1, and drops below 10° when f-number is larger than 2. Compared to [9], our system has slightly larger RMS phase error for larger f-number, because our system adopts wider pyramidal view compared to the narrower rectangular view in [9].

Next we study the effect of the quadratic approximation method outlined in Section 2.3.2. Use of quadratic approximations increases the phase error only mildly, as shown by the bold dashed line in Fig. 3a.

Fig. 3b shows the RMS phase error for different values of θ and ϕ . The error correlates with ϕ but remains almost the same for different θ , because the separable delay approximation method does not have any cross terms involving both x and ϕ . We see that the largest error appears in the four corners.

We next present point spread functions (PSF) for three different depths corresponding to f-numbers of 0.8, 1.4 and 2.0. We contrast results of separable and non-separable beamforming in Fig. 4. For separable beamforming, we use our iterative delay approximation. The simulation results show that even in the near field, our methods generate a nearly identical PSF as the non-separable method.

Finally, we show the simulation results for a 10-mm-diameter anechoic cylinder phantom at a depth corresponding to f-number of 1.4. Images of vertical and horizontal slices crossing the center of the cylinder shape cyst are also compared. The simulation results of separable and non-separable beamforming shown in Fig. 5 illustrate that the image quality for the two cases are indistinguishable.

To quantify the image quality, we evaluate contrast-to-noise ratios (CNR), as defined by equation (8).

$$\text{CNR} = \frac{|\mu_{\text{cyst}} - \mu_{\text{bgnd}}|}{\sqrt{\sigma_{\text{cyst}}^2 + \sigma_{\text{bgnd}}^2}} \quad (8)$$

where μ_{cyst} and μ_{bgnd} correspond to brightness of cyst and background, while σ_{cyst} and σ_{bgnd} are the standard deviation of cyst and background. In the image we use the area in the white boxes on the horizontal slice images for CNR evaluation. The non-separable beamforming image has a CNR of 5.0, while separable beamforming results in a CNR of 4.9. For a simulation configuration corresponding to f-number of 1.4, our image has slightly higher CNR compared to [9].

4. THE EVOLUTION OF SONIC MILLIP3DE FOR SEPARABLE 2-D ARRAY BEAMFORMING

Recently we proposed Sonic Millip3De, a front-end ultrasound architecture that combines 3-D die stacking techniques with a new beamforming pipeline to achieve massive parallelism with low power [2]. 3-D die stacking is an emerging IC fabrication technique that integrates several IC die layers on top of one another, such that wiring latency and power consumption are greatly reduced [12]. Fig. 6 depicts the 3-layered architecture, wherein layers are stacked vertically and connected with through silicon vias (TSVs). The first layer consists of CMUT transducers. We assume a 120×88 layout. The second layer consist of 12-bit ADCs and SRAM array. The third layer reads data from the SRAMs and performs the delay-sum beamforming operation using multiple independent pipelines to generate sub-images for different transducers in parallel. The beamforming pipelines each perform a linear interpolation on the data read from SRAM and feed the expanded data into “select” sub-units. These sub-units map the interpolated signal data into image data by iteratively calculating delay constants using piecewise quadratic approximations and selecting the appropriate signal value for the next focal point in the image. Multiple image segments are created in parallel by having several “select” sub-units on each pipeline, allowing a transducer’s entire sub-image to be generated over only a few iterations of the data. These sub-images are finally combined in the third layer’s network using adders to create a final image which is then stored in low-power DRAM. The microarchitecture of each “select” sub-unit is detailed in [2] and is unchanged for separable beamforming.

For non-separable beamforming, Sonic Millip3De combines sub-images by passing scanline data around a single 1,024-node ring network, accumulating the signal from “select” sub-units in a pipelined fashion. This simple pipelined summation is insufficient for separable beamforming, as the output of the first beamforming stage must be stored for processing in the second stage.

For separable beamforming, we extend Sonic Millip3De with additional SRAM storage on the second layer, which captures the output of the first-stage beamforming for each channel without overwriting the original echo data, and additional interconnect to handle the data flow of the separable beamforming algorithm. We organize the accelerator as 32 independent 32-node clusters. In the

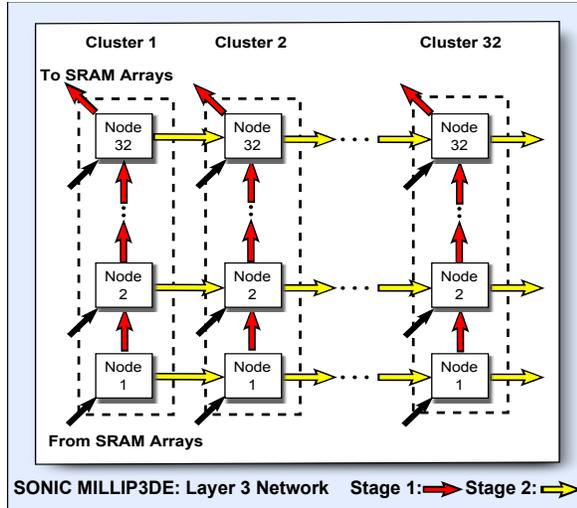


Fig. 7: Data flow in stages 1 & 2 of beamforming for Layer 3: The processing units in a dashed box form a cluster; the black arrows correspond to data flow from SRAM arrays to nodes in a cluster. In the 1st beamforming stage the data flows from bottom to top, in the 2nd beamforming stage the data flows from left to right. The output data from the second beamforming stage is stored in the DRAM.

first beamforming stage, scanline data is accumulated across the 32 nodes within a cluster and then written to SRAM, using the signal paths labeled “Stage 1” in Fig. 6b. Once the first beamforming stage is complete, its output is then passed through the accelerator a second time with constants for the second partial beamforming operation using the “Stage 2” paths shown in Fig. 6b, which run across the 32 clusters. Fig. 7 shows how data flows within a cluster during the first stage (data flows from bottom to top) and across clusters during the second stage (data flows from left to right). After completion of the computation of the second beamforming stage, the output image is stored in external DRAM. Reorganizing the accelerator into independent clusters has the added benefit of reducing the latency to load constants from off-chip memory during preprocessing, as the independent clusters can now load constants in parallel. The added SRAM and wiring result in a small ($< 2\%$) power increase.

By adding SRAM arrays and reorganizing the interconnect, we enable Sonic Millip3De to support separable beamforming with only minor modifications. Additionally, the throughput of the system is substantially increased by the new algorithm. With 1,024 nodes and 16 sub-units per node, this configuration is able to process all 48 scanlines for the 1,024 transducer sub-aperture in Stage 1 in only 3 iterations. Stage 2 requires 6 iterations (due to the 1,536 channels of data produced by Stage 1) giving a total of 9 iterations per sub-aperture. This is a $16\times$ improvement over the 144 iterations required by the non-separable baseline. We are also able to achieve additional power savings by power gating half of the system in the last 3 iterations of stage 2 as only 512 channels are left to process, enabling an energy savings of nearly $19\times$.

5. CONCLUSION

We propose a novel beamforming method that decomposes 2-D array beamforming into two stages of 1-D array beamforming. Our approach is based on a delay decomposition method that minimizes phase error. The resulting method reduces the beamforming

complexity of a SAU-based 3-D imaging system by $19\times$; the complexity reduction can be used to increase frame rate or decrease the power consumption of the hardware implementation. Field II simulation results show that this method can generate images with high CNR that are comparable to those generated by the non-separable beamforming method. Finally, we show that we can support separable beamforming on our recently proposed Sonic Millip3De by changing the beamforming network from a pipelined ring into a mesh topology to support the new data flow pattern and adding additional SRAM buffers to store intermediate data.

6. REFERENCES

- [1] S. Campbell, C. Lees, G. Moscoso, and P. Hall, “Ultrasound antenatal diagnosis of cleft palate by a new technique: the 3D reverse face view,” *Ultrasound in Obstetrics and Gynecology*, vol. 25, no. 1, pp. 12–18, Jan. 2005.
- [2] R. Sampson, M. Yang, S. Wei, C. Chakrabarti, and T. F. Wensich, “Sonic Millip3De: Massively parallel 3D stacked accelerator for 3D ultrasound,” *19th IEEE International Symposium on High Performance Computer Architecture*, pp. 318–329, Feb. 2013.
- [3] M. Karaman, H. S. Bilge, and M. O’Donnell, “Adaptive multi-element synthetic aperture imaging with motion and phase aberration correction,” *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 45, no. 4, pp. 1077–1087, July 1998.
- [4] G. R. Lockwood, J. R. Talman, and S. S. Brunke, “Real-time 3-D ultrasound imaging using sparse synthetic aperture beamforming,” *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 45, no. 4, pp. 980–988, July 1998.
- [5] R. Sampson, M. Yang, S. Wei, C. Chakrabarti, and T. F. Wensich, “Sonic Millip3De with Dynamic Receive Focusing and Apodization Optimization,” *IEEE Ultrasound Symposium*, July 2013.
- [6] M. Karaman, I. O. Wygant, O. Oralkan, and B. T. Khuri-Yakub, “Minimally redundant 2-D array designs for 3-D medical ultrasound imaging,” *IEEE Transactions on Medical Imaging*, vol. 28, no. 7, pp. 1051–1061, July 2009.
- [7] J. W. Choe, O. Oralkan, and P.T. Khuri-Yakub, “Design optimization for a 2-D sparse transducer array for 3-D ultrasound imaging,” *IEEE Ultrasound Symposium (IUS)*, pp. 1928–1931, Oct. 2010.
- [8] A. C. Dhanantwari, S. Stergiopoulos, L. Song, C. Parodi, F. Bertor, P. Pellegritti, and A. Questa, “An efficient 3D beamformer implementation for real-time 4D ultrasound systems deploying planar array probes,” *IEEE Ultrasound Symposium*, vol. 2, pp. 1421–1424, Aug. 2004.
- [9] K. Owen, M. I. Fuller, and J. A. Hossack, “Application of X-Y separable 2-D array beamforming for increased frame rate and energy efficiency in handheld devices,” *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 59, no. 7, pp. 1332–1343, July 2012.
- [10] J. A. Jensen, “FIELD: A program for simulating ultrasound systems,” *10th Nordicbaltic Conference on Biomedical Imaging, Vol. 4, Supplement 1, Part 1:351–353*, pp. 351–353, 1996.
- [11] J. A. Jensen and N. B. Svendsen, “Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers,” *IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control*, vol. 39, no. 2, pp. 262–267, Mar. 1992.
- [12] B. Black, M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G. H. Loh, D. McCaule, P. Morrow, D. W. Nelson, D. Pantuso, P. Reed, J. Rupley, S. Shankar, J. Shen, and C. Webb, “Die stacking (3D) microarchitecture,” *39th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO-39*, pp. 469–479, Dec. 2006.