Apodization Scheme for Hardware-Efficient Beamformer

A. Ibrahim*, F. Angiolini*†, M. Arditi†, J.-P. Thiran†‡, G. De Micheli*

*LSI, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

†LTS5, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

‡Department of Radiology, University Hospital Center (CHUV) and University of Lausanne (UNIL), Switzerland

***Abstract*—3D ultrasound is an emerging diagnostic technique that extends standard ultrasound imaging by capturing volumes, instead of planes. This brings completely new diagnostic opportunities, among which the possibility of disjoining image acquisition and analysis, thus enabling remote diagnosis, which would bring obvious medical and economic benefits.**

Unfortunately, 3D ultrasound is several orders of magnitude more computationally complex than 2D imaging. Therefore, algorithmic improvements to simplify the processing are mandatory in order to conceive cheap, portable, low-power imagers.

The kernel of the 3D imaging process, called beamforming, consists essentially of computing *delay* and *apodization profiles*. We have previously devised an approximation of the delay calculation stage, which dramatically reduces hardware complexity. Unfortunately, this approximation introduces an intrinsic degree of inaccuracy that can be characterized as added image noise.

In this paper, we identify an efficient approximated approach to the calculation of apodization profiles, that additionally minimizes (-76%) the error introduced during delay calculation. Together, these two techniques enable an efficient computation of 3D ultrasound images.

I. Introduction

Ultrasound imaging is a diagnostic technique among the most commonly used in medical practice, due to its comparative simplicity and non-invasiveness. Traditionally, it generates 2D images, corresponding to a plane orthogonal to the ultrasound probe face. This property means that the image acquisition and analysis are strongly coupled; a meaningful diagnosis can only be reached if the radiologist is also the probe operator, and is able to swipe and tilt the probe on the patient's body until the relevant anatomic feature is observed.

3D imaging is highly beneficial not only in diagnostic terms - where it dramatically improves, for example, any type of volumetric assessment - but also as it unlocks new imaging paradigms. The acquisition of a volume at once means that even an unskilled operator can position the probe with approximate precision, and still acquire the image plane of greatest interest. The imaged volume can then be examined by a remote specialist, e.g. in a faraway hospital. This creates exciting opportunities for diagnosis whenever it is infeasible for the specialist to be present in person, such as in rescue operations, in rural areas, in undeveloped regions. Savings of time and money could also be possible if family doctors could perform at least some type of scans in their offices or during home visits, relieving hospitals from the burden of long exam queues.

Unfortunately, all these applications require medical devices that must be transportable or even portable, cheap, and maybe also battery-operated for extended durations of time. This clashes with the fact that 3D ultrasound imaging is orders of magnitude more computationally expensive than conventional 2D imaging. Indeed, as of today, commercial 3D scanners are only available in hospital environments, and they are expensive, bulky and power-hungry [1], [2]. Research systems are also available, but expensive and power-hungry as well [3], [4].

These observations motivated us to search for computationally-efficient algorithms to tackle the digital processing, called *beamforming*, of image volumes. Beamforming consists of coherently summing up array-element echo signals received from anatomic features in the body; in this way, the location of acoustic impedance discontinuities - i.e. tissue boundaries - can be identified. Beamforming consists of three main operations: (i) the echo signals must be shifted in time depending on the image point under consideration, (ii) they must be weighted (*apodized*) to optimize antenna directivity, and (iii) they must be summed. The first two steps - calculation of delays and apodization - are the most challenging and must be tackled efficiently to enable a compact, cheap, low-power 3D imager.

In our previous work [5], [6], we tackled the problem of delay calculation, showing an approach that reduces dramatically the computation cost at the price of a degree of inaccuracy. This inaccuracy can be seen as resulting in added image noise. In this paper, we propose a solution to the apodization problem that, while maintaining the complexity reasonable, additionally compensates the inaccuracy introduced during delay calculation. We have conducted simulations of ultrasound fields (both emitted and pulse-echo) associated with a matrix probe using Field II [7], [8]. To the best of our knowledge, this approach is unique as it specifically matches our proposed delay calculation circuitry. Taken together, these two methods result in a cheap but high-quality architecture for low-power 3D imagers. This paper constitutes a stepping stone towards our final objective, which is to fit a high-performance beamformer onto a single FPGA.

II. Architecture

A. Beamforming and Delay Calculation

To beamform an image, either 2D or 3D, the echoes sampled at each transducer element must be summed according to a delay profile that models the propagation time necessary to travel from a body scatterer back towards each element of the probe. All modern systems apply dedicated delay profiles for each image point, i.e. they "focus" at every location as a function of the pulse-echo round-trip propagation time (*dynamic receive focusing*). Unfortunately, this severely increases the computation cost of image reconstruction.

To beamform each point S in the volume of interest V, this kernel must be computed $\forall S \in V$:

$$s(S) = \sum_{\forall D} e(D, t_p(O, S, D))w(D, S), \tag{1}$$

where the outcome $s(S)$ is a signal that represents the reflectivity of scatterers at S, and will eventually be used to calculate the brightness of the corresponding image pixel. e is the amplitude of the echo received by each probe element D at the time sample t_p. The value of t_p represents the propagation delay that sound waves incur from a given emission reference O, to the point S, and back to the probe's destination element D. Finally, the echo amplitude signals are weighted by $w(D, S)$, that represents apodization and will be discussed later.

In a 3D imaging setup, it is possible to gauge the complexity of solving this equation. A matrix transducer probe can have 10000 elements D (e.g. 9212 in [1]), while a volume of interest can consist of more than 16 million voxels (see Table I for reference system specifications). For a target frame rate of 15 fps, this means that delays must be calculated at a rate of 2.5 Tsamples/s. Each delay value is obtained from a geometric distance (transmit to S and back to D), which requires computing a square root. It is impractical or infeasible to solve the problem exactly with limited hardware resources. Even state-of-the-art hospital systems simplify the task by

978-1-5090-0493-5/16/$31.00 © 2016 IEEE

applying *pre-beamforming*, i.e. by summing together in the analog domain the signals from multiple probe elements onto a single signal. This can reduce the computational complexity by a large factor (e.g., about 40 times by coalescing 10000 probe element signals onto 256 cable wires) but also degrades the resolution advantages that such a transducer potentially offers.

In our previous papers [5], [6] we have proposed a significant algorithmic simplification that relies on a Taylor expansion of the square root calculation. This method approximates the delay to[1]:

$$t_p(O,S,D) \quad \approx \quad t_p(O,R,D) - \frac{x_D \sin\theta}{c} - \frac{y_D \sin\phi\cos\theta}{c} \qquad (2)$$

Where the expensive square root computation is replaced by two much cheaper subtractions. All terms on the right side of Equation 2 can be precomputed and efficiently compounded [5], [6].

Nonetheless, this approach is inaccurate; for some locations S, it introduces delay calculation errors of up to around 3 μs. As the delay values are used to select a sample of the received echoes, this error equates to picking echo signals almost 100 samples off [5], [6], and can essentially be characterized as a noise increase. We have further shown [6] that a way to get rid of this effect is, at those locations S, to ignore the element D echoes that are most impacted by the delay calculation inaccuracy. By discarding about 18% of the received echoes (Figure 3(a)), it was possible to reduce the delay calculation error to very low figures, typical of fixed-point designs that include rounding to an integer index for accessing the data sample array (maximum absolute error = 2.6749 delay samples, average = 0.6320). The discarding process is ruled by the observation that echoes that incur a delay calculation error of more than 2 samples (representing in our case a phase offset of 90°) go from providing constructive to destructive interference, and must thus be pruned.

This result however leaves open two issues. On one hand, it does not yet tackle the apodization ($w(D,S)$) problem; on the other, it does not explain precisely how the echoes that incur delay inaccuracies should be discarded. This paper tackles both topics at once.

B. Apodization

The weight coefficient $w(D,S)$ performs apodization [9], [10]. The main reason why apodization is needed is that the transducer behaves like an antenna array, and just like the latter, both its emission and reception diagrams exhibit *side lobes* according to the direction. In other words, the transducer is more sensitive in the intended direction (*main lobe*) but it also gathers echoes from other directions (*side lobes*), which contribute artifacts and noise to the image. The amplitude of the side lobes can be reduced by applying a weighting function to the received echoes across the array elements. Typical functions include Hanning, Hamming, and others [9]. These functions are usually smooth and gently taper off at the edges of the transducer.

The apodization function can be applied to the whole transducer, or only to a portion of it, consequently completely discarding the elements outside the apodization window. Normally, a concept called *expanding aperture* is used [10]; the applied apodization is very tight for locations S close to the transducer face, and is gradually expanded to cover the whole transducer (maximum aperture) for locations sufficiently far in front. This ensures, in principle, the best resolution and noise rejection across the volume.

TABLE I
SYSTEM SPECIFICATIONS

Parameter	Symbol	Value
Speed of sound in tissue	c	1540 m/s
Transducer center frequency	f_c	4 MHz
Transducer bandwidth	B	4 MHz
Transducer matrix size	$e_x \times e_y$	100 × 100
Wavelength	λ	c/f_c = 0.385 mm
Transducer element-pitch		$\lambda/2$
Matrix side dimension	d	50λ = 19.25 mm
Imaging volume ($\theta \times \phi \times d_p$)		73°×73°×500λ
Sampling frequency	f_s	32 MHz
Focal points		128 × 128 × 1000

C. Proposed Apodization for Efficient Beamforming

In this paper, we propose to apply the concept of expanding apodization in such a way that it compensates for delay inaccuracy by discarding certain echoes, but at the same time keeping its implementation as simple as possible with an FPGA target in mind. We do so by way of the following:

1) We examine the volume and derive, for each voxel, maps of (i) the usual expanding apodization window (Figure 1, white rectangle), and (ii) the set of elements that incur too large delay inaccuracies (Figure 1, region outside of the magenta contour).
2) We manually observe several of these maps, measuring the size of a new rectangular apodization window, subset of the base one, that discards as many "inaccurate" elements as possible while keeping as many of the "accurate" as possible (Figure 1, black rectangle).
3) We build by regression an initial set of equations that define the apodization windows (height, width, center abscissa, center ordinate) for varying locations S in the volume. This yields four equations as a function of r_S, θ_S, ϕ_S.
4) We run a script that exhaustively analyzes the volume to drive refinements of these equations, in order to trim or expand the apodization windows to achieve better results. The results are shown in Section III.

It is worth noting that this is an exercise in tradeoffs, for several reasons. First of all, the apodization windows should ideally be unique for every S, but this is completely impractical - storing 16 million apodization windows, each defined over 100 by 100 elements, requires too much storage. It is similarly impractical to construct these windows in realtime, as they usually require computing a complex expression (e.g., a 2D Hanning) at an extremely fast rate. Therefore, a limited set of apodization windows should be precomputed and reused.

Additionally, the profile of the geometric inaccuracy (Figure 1) is such that to precisely discard the "inaccurate" elements and only those, a complex geometric pattern should be defined, and an even more complex smoothly degrading function on top. To simplify the implementation, we choose to only work with 2D Hanning functions defined on rectangles; we must then trim profiles such as that of Figure 1 to a rectangle. This unavoidably either rejects a few usable elements or accepts a few "inaccurate" ones, which requires a balance. The results of this effort are shown in Section III.

At runtime, the FPGA circuitry can easily solve our derived equations for each voxel, thus devising the size and position of the required apodization window. The windowing function, i.e. the 2D Hanning, can then be fetched from an on-FPGA memory. In the worst case, the total number of required apodization windows, without considering translations, is 10000 - from the possible size of a rectangle that fits in a 100 by 100 square. This number is reduced to 5000 considering X-Y symmetries. If each window is stored as a rectangle comprising 100×100 values, defined at a precision of e.g. 8 bits, a total size of 50 MB is required. This number can be reduced to 13 MB if each window is compactly stored at its intrinsic size (e.g. 1×1, 1×2, ...), at the cost of more addressing logic. This number is challenging, but would fit, for example, in a single chip of the latest Xilinx Ultrascale+ family [11], which sports up to 12 MB of BRAM plus 45 MB of UltraRAM. We also study the accuracy of a further simplification that reduces storage dramatically, namely that of only using square apodization windows, which reduces storage size to 1 MB, or respectively 323 kB if storing the windows in a compact format, the same as needed by a plain expanding aperture design.

III. Experimental Results

After multiple refinements, we derived equations like that of Figure 2. These are the width, height, and the center - x_c and y_c - of the apodization window for a single zone of the volume. As can be seen, the width and the height are expressed as a function of

[1] The expression depends on whether the volume is swept azimuth-first or elevation-first.

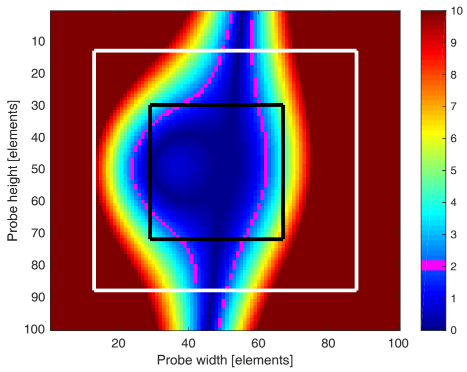


Fig. 1. Inaccuracy of the delay calculation towards each element of the probe; the chart represents the probe face, and the color shade describes the calculation inaccuracy at that element D expressed in sampling periods, from a voxel S at ($r = 8mm$, $\theta = -28°$, $\phi = -3°$). Blue = error of less than 1 delay sample, red = error of 8 samples. The contour line (magenta) defines the profile of the elements incurring an inaccuracy of $\leq$2 samples, i.e. the split between constructive and destructive contribution. White rectangle: standard expanding apodization window; black rectangle: apodization window derived in this work. Note how the latter window tracks more closely the 2-sample contour.

r, θ, ϕ, while x_c is expressed as a function of r, θ, and y_c is expressed as a function of r, ϕ. As the size and the location of this window fluctuate in very complex ways, we have chosen to use different equations to model them in different sections of the volume, although the modeling cannot be perfect. These windows are a smaller version of the usual expanding apodization windows, and thus completely replace them e.g. for the purpose of ignoring elements that should be excluded due to limited directivity. We omit the rest of the equations that describe the other sections of the volume for space reasons.

$$W(r,\phi,\theta) = 0.5(\frac{13r}{140} - \frac{\theta+37}{3.5} - \frac{\phi+37}{3.5} + 19) \ , \quad (140 \leq r < 1500) \wedge (-36° \leq \theta \leq -19°) \wedge (-18° \leq \phi \leq 0°)$$

$$H(r,\phi,\theta) = 0.5(\frac{13r}{140} + \frac{\theta+37}{3.5} + \frac{\phi+37}{3.5}) \ , \quad (140 \leq r < 1500) \wedge (-36° \leq \theta \leq -19°) \wedge (-18° \leq \phi \leq 0°)$$

$$x_c(r,\theta) = 49 - \frac{3r}{420} + \frac{\theta+37}{10.5} \ , \quad (140 \leq r < 1500) \wedge (-36° \leq \theta \leq -19°)$$

$$y_c(r,\phi) = \begin{cases} 49 - \frac{3r}{420} + \frac{\phi+37}{10.5} \ , & (140 \leq r < 1500) \wedge (-18° \leq \phi \leq -4°) \\ 51 \ , \quad (140 \leq r < 1500) \wedge (-3° \leq \phi \leq 0°) & \end{cases}$$

Fig. 2. Equations expressing the width, the height, and the center (x_c and y_c) of the apodization window. The angle variables are expressed in degrees and the r variable is expressed in periods of sampling frequency. The equation holds in one region of the volume and is replicated symmetrically in theta, phi.

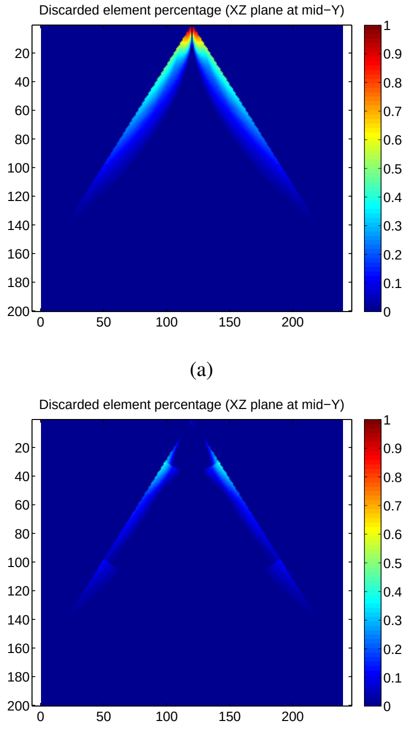
The trimmed windows defined by these equations enable a large improvement in the rejection of elements that are subject to high inaccuracy of delay calculation, as can be seen in Table II. This can also be seen graphically in Figure 3. We could achieve a reduction from 15.2% to 3.6% (-76%) of the elements subject to large delay inaccuracy, with a reduction of less than 30% in the number of elements used for beamforming. The memory requirement is 1.51 MB, about ten times lower than the worst case of 13 MB estimated in Section II; this is because we observed that a large number of permutations of window sizes were not used by the model.

Next, we also evaluated the opportunity of defining the apodization windows as square, in order to save memory. This reduces the set of equations required to only three, one for both width and height, and two for the centering. Fig. 4 shows that trimming the apodization improves slightly the picture quality in the main area of the image. The main improvements however are at the edges; a more detailed illustration is not possible for space reasons and will be provided in full in future publications. The outcomes in terms of precision and memory requirements are also shown in Table II. We discovered that in return for a negligible loss in rejection, it was possible to dramatically decrease the required amount of memory storage, bringing it back in line with the baseline of a standard expanding aperture. We therefore consider this to be the optimum design point.

	Elements Apodized Away	**Elements Kept (Accurate Delay)**	**Elements Kept (Inaccurate Delay)**	**Memory**
Expanding Apodization	2.5%	82.3%	15.2%	323 kB
Trimmed	38.2%	58.1%	3.6%	1.51 MB
Trimmed, square	38.3%	58.0%	3.7%	323 kB

TABLE II

COUNT OF ELEMENTS THAT ARE APODIZED AWAY BY DIFFERENT APODIZATION FUNCTIONS; FOR THE REMAINING ELEMENTS, COUNT OF THE ELEMENTS THAT EXPERIENCE ACCURATE OR INACCURATE DELAY CALCULATION (THRESHOLD = 2 SAMPLING PERIODS OF INACCURACY). MEMORY BUFFER SIZE REQUIRED TO STORE THE CORRESPONDING APODIZATION FUNCTIONS, E.G. 2D HANNING FUNCTIONS.



(a)

(b)

Fig. 3. The inaccuracy in delay calculation becomes negligible by discarding all the echoes from elements that incur more than 2 samples of calculation inaccuracy. These plots show, for each voxel S on the XZ plane of the image, the percentage of such elements when using (a) a standard expanding-aperture apodization, (b) a tighter apodization as proposed in this paper. A significant inaccuracy reduction can be noticed. The remaining inaccuracy is confined to the edges of the image, which are clinically less essential.

IV. Previous Work

Today's state-of-the-art 3D ultrasound systems do analog beamforming in element subgroups in the transducer head to decrease the number of channels that are carried along the cable from a few thousands to a few hundred [13]. This is called "pre- or micro-beamforming", where precomputed fixed analog delays are applied to the signals received by groups of transducer elements, and compounded in a single analog signal. The pre-beamforming output, which has much lower bandwidths, is then used for digital beamforming. Of course this results in a loss of potential image quality, as the signal count must be reduced by up to 40:1.

The problem of how to compute delay coefficients to feed high-channel-count beamformers at a very high throughput has been recognized as critical. For example, Sonic Millip3De [14] implements

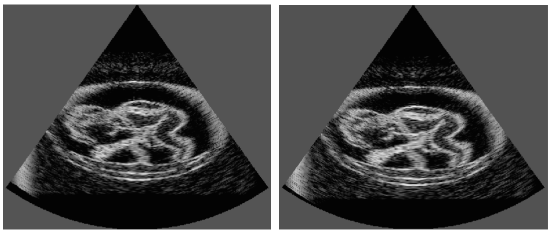


Fig. 4. 3D imaging of a fetus phantom provided online through [12] using (a) expanding apodization, or (b) trimmed square apodization.

ultra-fast imaging for 128×96 transducer elements (of which only 1024 are considered per shot) with a powerful die-stacked package. Its main bottleneck is that it requires a large external DRAM memory to store beamforming delay coefficients, and a very high memory bandwidth. Other works [15], [16], [17], [18] have shown that a feasible alternative is to try to compute all delay coefficients on-the-fly on-chip. Since this computation involves the evaluation of complex functions like square roots, it is mandatory to identify accurate, fast and low-area approximation circuits [19], [20]. In particular, we have shown [5], [6] how a highly efficient architecture can be devised by accepting some inaccuracy in the calculation of the delays.

The kernel of ultrasound beamforming includes also the step of apodization. Apodization is a well-known signal processing step used in multiple fields, for example in the design of electromagnetic antennas, where it helps shaping radio beams. Similarly, apodization can improve ultrasound image quality by reducing the amplitude of the side lobes of the ultrasonic beam [9], and thus improving the cleanliness and contrast. Unfortunately, this improvement comes at a resolution cost, due to the widening of the main lobe, and therefore the most commonly used apodization windows, e.g. Hanning, must strike a balance between these two metrics [9].

Apodization can be applied to the transmitted beam, but it is even more critical during receive. As typical ultrasound systems focus dynamically on the voxels of the volume, calculating the corresponding delay profiles on-the-fly as discussed above, the apodization function must be, in principle, tuned differently for each such point. This problem is very hard to solve analytically, and more so to solve in a computationally-efficient way, although attempts have been made for 2D imaging [21], [22], [10]. Drastic simplifications have been adopted in the few papers discussing apodization for 3D ultrasound [14].

In this paper, we propose to use a custom apodization function to mask the inaccuracies of an approximate architecture for delay computation. To the best of our knowledge, the requirements and implementation of this function are unique and novel. Further, we tune the function to have a feasible implementation on FPGA. This is to achieve our ultimate goal of a first complete high-quality and low-power 3D ultrasound back-end system that suits portable usage scenarios.

V. Conclusions

3D ultrasound imaging systems open new opportunities for remote diagnosis, with enormous potential benefits in medical and economic terms. Unfortunately, to fully realize those benefits, it is also necessary to build imagers that are portable, cheap and battery-operated. This is difficult as 3D ultrasound imaging requires a massive amount of calculations.

In previous papers, we had demonstrated major circuit complexity savings by approximating the calculation of propagation delays. Unfortunately, that method introduces an inaccuracy that equates to image noise. In this work, we have both expanded our approach to the problem - adding apodization computation - and used the apodization architecture to cancel out (-76%) the previous inaccuracy. We have shown a technique that, while requiring an amount of storage equal to the baseline design, can effectively apply apodization as required by the imager.

Next, we plan on showcasing both the delay and the apodization computation architectures in a high-end FPGA demonstrator, proving the feasibility of a single-chip, high-performance 3D imager. We also plan on generalizing this work's conclusions to probes with different geometries and center frequencies.

Acknowledgment

The authors would like to acknowledge funding from the Swiss Confederation through the UltrasoundToGo project of the Nano-Tera.ch initiative.

References

[1] Philips Electronics N.V., "Philips iU22 ultrasound with xMATRIX system specifications," 2012, www.healthcare.philips.com.

[2] GE Healthcare, "Voluson E10," www.gehealthcare.com.

[3] J. Jensen, H. Holten-Lund, R. Nielson, B. Tomov, M. Stuart, S. Nikolov, M. Hansen, and U. Larsen, "Performance of SARUS: A synthetic aperture real-time ultrasound system," in *IEEE Symposium on Ultrasonics*, Oct. 2010, pp. 305–309.

[4] Verasonics, "Vantage 256," verasonics.com/vantage-systems/.

[5] A. Ibrahim, P. A. Hager, A. Bartolini, F. Angiolini, M. Arditi, L. Benini, and G. De Micheli, "Tackling the bottleneck of delay tables in 3d ultrasound imaging," in *Proceedings of the 2015 Design Automation and Test in Europe (DATE 2015) Conference*, March 2015, pp. 1683 – 1688.

[6] "Paper currently in review."

[7] J. Jensen, "Field: A program for simulating ultrasound systems," in *10th Nordic-Baltic Conference on Biomedical Imaging Published in Medical and Biological Engineering and Computing*, vol. 34, Supplement 1, 1996, pp. 351–353.

[8] J. Jensen and N. Svendsen, "Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers," *Ultrasonics, Ferroelectrics, and Frequency Control, IEEE Transactions on*, vol. 39, no. 2, pp. 262–267, March 1992.

[9] P. 't Hoen, "Aperture apodization to reduce the off-axis intensity of the pulsed-mode directivity function of linear arrays," *Ultrasonics*, vol. 20, no. 5, pp. 231 – 236, September 1982.

[10] C. M. W. Daft and W. E. Engeler, "Windowing of wide-band ultrasound transducers," in *Ultrasonics Symposium, 1996. Proceedings., 1996 IEEE*, vol. 2, Nov 1996, pp. 1541–1544 vol.2.

[11] Xilinx Inc., "Ultrascale architecture," 2014, www.xilinx.com/products/technology/ultrascale.html.

[12] "Field ii simulation program: Calculation of b-mode image of synthetic fetus," Field II, April 2012. [Online]. Available: http://field-ii.dk/?examples/fetus_example/fetus_example.html

[13] B. Savord and R. Solomon, "Fully sampled matrix transducer for real time 3D ultrasonic imaging," in *Ultrasonics, 2003 IEEE Symposium on*, vol. 1. IEEE, 2003, pp. 945–953.

[14] R. Sampson, M. Yang, S. Wei, C. Chakrabarti, and T. Wenisch, "Sonic millip3de: A massively parallel 3d-stacked accelerator for 3d ultrasound," in *High Performance Computer Architecture (HPCA2013), 2013 IEEE 19th International Symposium on*, Feb 2013, pp. 318–329.

[15] S. I. Nikolov, J. A. Jensen, and B. Tomov, "Recursive delay calculation unit for parametric beamformer," in *Medical Imaging*. International Society for Optics and Photonics, 2006, pp. 61 470D–61 470D.

[16] J. Jensen, H. Holten-Lund, R. Nilsson, M. Hansen, U. Larsen, R. Domsten, B. Tomov, M. Stuart, S. Nikolov, M. Pihl, Y. Du, J. Rasmussen, and M. Rasmussen, "Sarus: A synthetic aperture real-time ultrasound system," *Ultrasonics, Ferroelectrics, and Frequency Control, IEEE Transactions on*, vol. 60, no. 9, pp. 1838–1852, Sep 2013.

[17] P. Vogel, A. Bartolini, and L. Benini, "Efficient parallel beamforming for 3D ultrasound imaging," in *Proceedings of the 24th Edition of the Great Lakes Symposium on VLSI*, ser. GLSVLSI '14. New York, NY, USA: ACM, 2014, pp. 175–180. [Online]. Available: http://doi.acm.org/10.1145/2591513.2591599

[18] Cephasonics Receive RX Beamformer, CSC2032, http://www.cephasonics.com.

[19] S. Nikolov, J. Jensen, and B. Tomov, "Fast parametric beamformer for synthetic aperture imaging," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control.*, vol. 55, no. 8, pp. 1755–1767, August 2008.

[20] J. Park, J. Lee, D.-H. Kim, M. Kim, J. H. Chang, T.-K. Song, and Y. Yoo, "Efficient implementation of a real-time dynamic synthetic aperture beamformer," in *IEEE Symposium on Ultrasonics*, Oct 2012, pp. 2250–2253.

[21] J. Bhattacharyya, P. Mandal, R. Banerjee, and S. Banerjee, "Real time dynamic receive apodization for an ultrasound imaging system," in *VLSI Design, 2006. Held jointly with 5th International Conference on Embedded Systems and Design., 19th International Conference on*, Jan 2006, pp. 4 pp.–.

[22] B. Tomov and J. Jensen, "Compact implementation of dynamic receive apodization in ultrasound scanners," in *Medical Imaging 2004, in Progress in Biomedical Optics and Imaging - Proceedings of SPIE*, Feb 2004, pp. 260–271.