Architectural Support for Medical Imaging

by

Richard A. Sampson II

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in The University of Michigan
2017

Doctoral Committee:

Associate Professor Thomas F. Wenisch, Chair
Professor Jeffrey A. Fessler
Research Associate Professor Oliver D. Kripfgans
Professor Scott Mahlke

Richard A. Sampson II

rsamp@umich.edu

ORCID iD: 0000-0002-8669-6584

For William, my grandfather, and for William, my son.

ACKNOWLEDGEMENTS

This thesis is the result of many years of hard work that simply would not have been possible without the continuous support of so many. Their encouragement, patience, and guidance are the heart of this work and are the reason I was able to persevere.

I would like to first thank my adviser, Thomas Wenisch, whose passion and knowledge always drove me to improve myself and my research. I would also like to thank my committee members, Jeffrey Fessler, Scott Mahlke, and Oliver Kripfgans, as well as my numerous collaborators, whose advice and feedback were invaluable to my research and this thesis. I want to thank my friends and my family, especially those who rarely see me and have no idea what I do, but give me their love and support anyway. Finally, I want to thank my wife, Christina, who encouraged me, supported me, and shared all of my burdens and hardships to help me get to the end.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

Architectural Support for Medical Imaging

by

Richard A. Sampson II

Chair: Thomas F. Wenisch

Advancements in medical imaging research are continuously providing doctors with better diagnostic information, removing the need for unnecessary surgeries and increasing accuracy in predicting life-threatening conditions. However, newly developed techniques are currently limited by the capabilities of existing computer hardware, restricting them to expensive, custom-designed machines that only the largest hospital systems can afford or even worse, precluding them entirely. Many of these issues are due to existing hardware being ill-suited for these types of algorithms and not designed with medical imaging in mind.

In this thesis we discuss our efforts to motivate and democratize architectural support for advanced medical imaging tasks with MIRAQLE, a medical image reconstruction benchmark suite. In particular, MIRAQLE focuses on advanced image reconstruction techniques for 3D ultrasound, low-dose X-ray CT, and dynamic MRI. For each imaging modality we provide a detailed background and parallel implementations to enable future hardware development. In addition to providing baseline algorithms for these workloads, we also develop a unique analysis tool that provides image quality feedback for each simulation. This allows hardware designers to

explore acceptable image quality trade-offs in algorithm-hardware co-design, potentially allowing for even more efficient solutions than hardware innovations alone could provide.

We also motivate the need for such tools by discussing Sonic Millip3De, our low-power, highly parallel hardware for 3D ultrasound. Using Sonic Millip3De, we illustrate the orders-of-magnitude power efficiency improvement that better medical imaging hardware can provide, especially when developed with a hardware-software co-design. We also show validation of the design using a scaled-down FPGA proof-of-concept and discuss our further refinement of the hardware to support a wider range of applications and produce higher frame rates. Overall, with this thesis we hope to enable application specific hardware support for the critical medical imaging tasks in MIRAQLE to make them practical for wide clinical use.

CHAPTER I

Introduction

Medical imaging has been a cornerstone of modern medicine since the advent of X-ray imaging over a century ago. Since then, there have been numerous advances from the introduction of diagnostic ultrasound in the 1940s to the development of magnetic resonance imaging (MRI) in the 1970s. Today doctors are able to easily and safely peer inside the body to examine everything from the brain to an unborn fetus without having to make a single cut. However, as these imaging techniques continue to be refined and become more advanced so do they become more computationally complex. This increased complexity demands more specialized hardware support; otherwise, the deployment of advanced imaging techniques and applications, especially for clinical use, will continue to be limited.

Currently several state-of-the-art imaging algorithms have been developed that can provide greater detail [67, 89], reduce radiation dosage [32, 50], or even enable increased real-time and mobile imaging applications [19, 63, 74, 95]. However, only high-end commercial medical imaging systems, which often use custom hardware solutions developed internally and can cost up to 1 million dollars each [1], have the capabilities to provide even partial support of these newly developed methods. Additionally such systems can only be afforded by large hospital systems, preventing many patients and doctors even potential access to these benefits. At the other end of

the spectrum are more widely available and cheaper commercial systems that rely on stock hardware solutions such as GPUs or servers that are not designed with medical imaging in mind and are inefficient in power and performance when used for such tasks. These commonly used systems often have severely reduced capabilities, and can provide only the most basic imaging techniques.

Many of these limitations in current medical imaging hardware stem from an unfamiliarity with these applications among the computer architecture community, leading to limited architecture research in this area and to hardware that is, instead, mostly ill-suited for this application space. However, research has shown that medical imaging specific solutions can provide orders-of-magnitude improvement in performance and efficiency [18, 19, 74], and that even simple improvements to existing hardware such as increasing bandwidth can boost performance substantially for applications where this is a major performance bottleneck [78]. Additionally with the ever increasing move to heterogeneous hardware and accelerators on chip [3, 13], there is an even greater opportunity for architects to provide focused support for these algorithms in computer hardware design to achieve these benefits and enable increased clinical use.

In this work we first discuss our efforts to improve medical imaging with Sonic Millip3De, a specialized full-system hardware design for 3D ultrasound. By co-designing both the hardware and underlying beamforming algorithm, we create a design that is orders-of-magnitude more efficient than existing hardware and is able to process up to 30% more channels than current commercial systems [66] and a real-time output of 1 frame per second which is on par with existing 3D systems. We also discuss our work in developing an FPGA proof-of-concept prototype to test our hardware and verify our simulation results. This design, while only a single channel of the fully proposed ASIC, achieves the expected image quality, and validates the expected performance of the full Sonic Millip3De system.

We also describe follow-up work where we further refine the Sonic Millip3De design to achieve an even higher 32 volumes per second frame rate while maintaining the same power requirement. This improvement is achieved through further algorithmic and hardware modifications to support a 2-step separable beamforming process developed by our collaborators at Arizona State University. Finally we demonstrate the flexibility of the system by showing that the hardware can even achieve over a 1k volume per second rate with acceptable image quality to enable ultra high frame rate motion estimation applications. Such estimation algorithms are currently limited to 2D but enabling 3D could expand the capabilities of a wide range of medical applications such as diagnosing life-threatening cardiac conditions [88], at-risk pregnancies [68], and chronic pulmonary issues [72, 73].

After discussing our previous work in ultrasound hardware, we introduce MIRAQLE[1], a medical image reconstruction benchmark suite to further bridge the gap between architects and medical imaging and enable the development of improved medical imaging hardware. In particular, we focus on the three state-of-the-art medical imaging tasks: 3D ultrasound beamforming, low-dose X-ray computed tomography (CT), and dynamic MRI. With MIRAQLE, we not only provide representative state-of-the-art algorithms and data sets for each imaging problem but also an image quality analysis tool to analyze output variation from any necessary algorithmic modifications. Our Sonic Millip3De work has shown that the greatest performance is achieved through hardware and algorithm co-design. Therefore, a primary goal of MIRAQLE is to enable architects to make algorithmic modifications that may result in acceptable image quality variation while providing additional performance gain.

[1]Available at miraqle.eecs.umich.edu

1.1 Need for Improved Architectural Support

Three-dimensional (3D) medical imaging has had a transformative impact on medicine. 3D and dynamic imaging provide diagnostic capabilities that are unavailable with 2D imaging and can improve diagnostic efficiency and patient throughput (e.g., [7]). Enhancing physicians' diagnostic capabilities mandates continued improvements in imaging resolution and quality.

Across imaging modalities, however, the signal-to-noise ratio of the acquired imaging data is inherently limited. In dynamic magnetic resonance imaging (MRI), limits are imposed by physics: the nuclear spin relaxation time of Hydrogen-1 isotopes imposes constraints on spatial and temporal resolution [26, 91]. In X-ray computed tomography (CT), concerns over radiation dose motivate persistent efforts to reduce X-ray source intensity without compromising image quality [60]. In ultrasound, physical constraints arise due to transmit bandwidth and attenuation at depth.

To advance imaging capability despite SNR constraints, bioimaging engineers have leveraged the bounty of Moore's Law, using ever more sophisticated and demanding computational techniques to improve imaging capability. Such computational techniques drastically improve image quality when compared to textbook image formation techniques by introducing *a priori* knowledge of imaging system physics, measurement noise statistics, and/or anatomy to improve image quality. For example, in 2011, the FDA approved GE's Veo method for low-dose X-ray CT reconstruction, the first commercial use of model-based iterative reconstruction (MBIR) to reduce radiation dose [29].

However, quality improvements obtained by such techniques come at a cost of three to five orders of magnitude increase in the computational burden. Therefore, the steady march of Moore's Law has been a central enabler for advanced image reconstruction. But, with the end of Moore's Law [11, 31] and Dennard Scaling [23, 38], bioimaging can no longer rely on inevitable annual improvement of CPU

and GPU capability. At the same time, the data sizes in medical imaging systems continue to grow (at a rate outpacing Moore's law) [9]. A new scaling strategy is needed.

In our initial work [74], our goal was to develop a three-dimensional (3D) ultrasound device with a power budget of a hand-held device. 3D ultrasound is a particularly attractive modality for hand-held imaging because ultrasound transducers use little power (limited by FDA regulations to a few hundred milliwatts [61]) and pose no known dangers or side-effects, in contrast to X-ray and MRI [69, 82]. 3D ultrasound provides numerous benefits over its 2D counterpart. Not only are 3D images easier to interpret, reducing effort (and errors) for technicians to locate relevant anatomy, they also provide accurate volumetric measurements of cysts and tumors that 2D cannot match. In fact, prior to 3D imaging, technicians sometimes resorted to estimating cyst volumes by mentally piecing together 2D slice images [14].

However, the benefits of 3D also come with numerous hardware challenges that are only exacerbated when trying make the system hand-held. To construct a 3D volumetric image, a conventional linear transducer array (e.g., 128 elements) for 2D imaging must be replaced with a rectangular array (128×96 in our aggressive design), increasing the incoming data rate by 100×. Furthermore, rather than reconstruct a typical 2D image resolution of 50×4096 focal points, the 3D image comprises 50×50×4096 focal points, another factor of 50 increase. The computational requirements increase by the product of these factors (nearly 5000×).

Because it is in close contact with human skin, an ultrasound scan head must operate within a tight power budget (about 5W) to maintain safe temperatures. Though transducer power is negligible relative to this limit, the raw data rate produced by a 128×96 high-resolution transducer array exceeds 6 Tb/s—so high that it cannot even be transferred off chip for processing. In 2D systems, delay constants used for beamforming are easily pre-computed and stored; for our target 3D system,

over 125 billion such constants are required and must be computed on-the-fly, nominally requiring billions of square root and trigonometric operations. The challenge of 3D hand-held ultrasound lies in performing these computations within a 5W budget. Implementing 3D ultrasound with commercially available DSP/GPU chips and conventional beamformation algorithms is simply infeasible, requiring over 700 DSP chips with a total power budget of 7.1kW.

To meet this challenge, we develop Sonic Millip3De, a system architecture and specialized accelerator unit for low-power 3D ultrasound beamformation. The Sonic Millip3De makes use of a massively parallel hardware design and state-of-the-art 3D die stacking [8, 28, 49, 53], splitting analog components, analog-to-digital (ADC) converters and SRAM storage, and a 1024-unit beamsum accelerator array across three silicon layers for a compact design with short (and hence low-power) wires. The accelerator array is organized according to a streaming design paradigm and is enabled by a novel algorithm for iteratively computing beamformation delay constants that balances pre-computed value storage with on-the-fly calculations while requiring only table look-up and add operations. The system architecture builds on recent ultrasound advances including sub-aperture multiplexing [37, 44] and virtual sources [45, 65].

Based on RTL-level design and floorplanning for an industrial 45nm process, we estimate a full-system power requirement of 16W for Sonic Millip3De and project that it will meet the 5W target power budget by the 16nm node, and we verify its performance using a scaled-down FPGA implementation of the RTL-level code. With this design we are able to achieve the multiple orders-of-magnitude improvement in performance over existing hardware. These improvements are the product of clever design as well as a hardware-software co-design, which leverage modifications to the algorithm itself to enable highly efficient hardware, and illustrate the need for better hardware support to close the performance gap need for state-of-the-art imaging

algorithms.

1.2 Medical Imaging Benchmark Suite

Specialized computer architectures hold enormous promise to scale imaging system performance into the next decade. Computing systems researchers have already demonstrated several successes with specialized architectural or GPU support for medical image reconstruction (e.g., [12, 19, 35, 39, 48, 51, 58, 74, 77, 78, 92]), but further advancement is critically needed. Approximate computing techniques (e.g., [24, 34, 36, 81, 85, 86]) hold particular promise for medical imaging applications as the acquisition and digitization of the source signal is inherently noisy. So, image reconstruction algorithms are already designed to tolerate and minimize noise. However, image quality trade-offs are inherent in architectural approximation and specialization techniques; improvements in computational performance must be coupled with careful evaluation of image quality and its impact on representative imaging tasks. The cross-disciplinary nature of this problem poses a challenge: expertise in computer architectures and medical imaging are typically disjoint.

For these reasons, we present MIRAQLE, a 3D medical image reconstruction benchmark suite and evaluation framework, to bridge this expertise gap. MIRAQLE seeks to *democratize medical image reconstruction research*; to make it possible for experts in computer architecture to study state-of-the-art image reconstruction algorithms in the context of representative imaging tasks and data sets and make quantitative computational performance vs. imaging quality trade-offs.

MIRAQLE comprises three imaging tasks: (1) synthetic aperture ultrasonic 3D brightness-mode (B-mode) imaging of lesions in tissue, (2) low-dose helical chest X-ray CT reconstruction, and (3) dynamic contrast-enhanced MRI of a brain with a tumor. For each task, we have created a parallel C implementation of a state-of-the-art image reconstruction algorithm, along with a reference input and output.

The key novelty of MIRAQLE is that it also includes automated tools to quantify a modality-specific quality metric for each imaging task. We have developed these metrics in consultation with experts in each imaging modality and in accordance with the typical evaluation practices in the relevant biomedical imaging literature. Using these tools, and guidance on quality thresholds we provide, researchers with limited background in medical imaging will still be able to assess quantitatively whether an approximate hardware implementation has an acceptable impact on image quality.

1.3 Outline

The goals of this work are to show the performance and efficiency possible through specialized architectural support, determine the key inefficiencies in existing hardware for state-of-the-art medical imaging algorithms, and present an easy-to-use framework for developing improved hardware to support. To accomplish this, we have organized the rest of this document as follows: Chapter II provides a high-level overview of ultrasound imaging and introduces Sonic Millip3De, our efficient hardware for 3D ultrasound imaging. We use this work to illustrate the potential orders-of-magnitude performance gain in efficient hardware design. In Chapter III we cover further improvements to the Sonic Millip3De design, allowing the hardware to achieve higher frame rates for real-time imaging as well as 1k frame rates to enable velocity and flow estimation without requiring additional system power. After concluding our discussion on Sonic Millip3De, we will provide an overview on X-ray CT and MRI in Chapter IV. Then in Chapter V, we outline MIRAQLE, our proposed benchmark suite. In this chapter we give an overview of the imaging algorithms, the key image quality metrics, and architectural opportunities for each modality. We also show results from a demonstration of our image quality evaluation which explores potential algorithmic changes that would simplify hardware. Finally we will conclude with Chapter VI where we discuss our final thoughts and directions for future research.

CHAPTER II

Sonic Millip3De

In this chapter we introduce Sonic Millip3De, an application specific hardware accelerator targeting low-power 3D ultrasound. The contents of this chapter are primarily from our work in [74, 75, 76, 77]. In the first section we give an overview of ultrasound imaging, focusing on the synthetic aperture delay and sum algorithm that is the basis for the Sonic Millip3De beamforming method. After that, we cover the algorithmic and hardware innovations that allow Sonic Millip3De to achieve high performance. Finally we present results demonstrating Sonic Millip3De's image quality and system performance, both from simulations as well as demonstrated by an FPGA prototype.

2.1 Ultrasound Imaging Background

This section gives an introduction into the basics of ultrasound, specifically focusing on synthetic aperture beamforming and other techniques commercial systems use to enable 3D ultrasound.

2.1.1 Synthetic Aperture Ultrasound Overview

Ultrasound is performed by sending high frequency pulses (typically 1-15MHz) into a medium and constructing an image from the pulse signals that are reflected

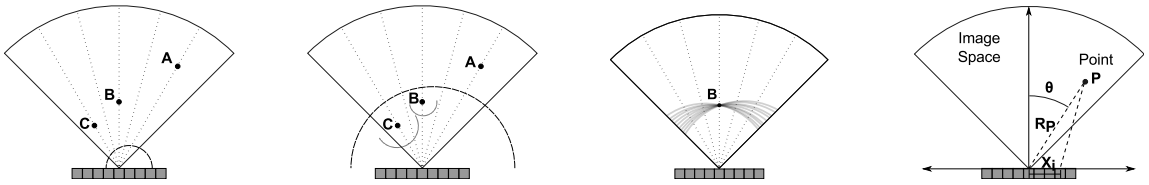


Figure 2.1: **Ultrasound Background.** (a) Pulse leaving transmit transducer. (b) Echo pulses reflecting from points B and C. All transducers in array (or sub-aperture) will receive the echo data, but at different times due to different round trip distances. (c) All of the reconstructed data for point B from each of the transducers added together. By adding thousands of "views" together, crisp points become visible. (d) Variables used in calculating round trip distance, d_p, for the i-th transducer and point P in Eq. 5.1.

back. The process comprises three stages (transmit, receive, and beamsum / beamforming) which can performed via several techniques. For this work, we will be focusing on synthetic aperture which uses an unfocused transmit and data is "focused on receive" during the beamforming stage via a delay and sum method. We use this technique in particular because other methods (such as focused transmit) require too many transmissions for 3D ultrasound to be useful for many applications.

Transmission and reception for 3D ultrasound are both done using an 2D array of capacitive micromachined ultrasonic transducers (CMUTs) that are electrically stimulated to produce the outgoing signal and generate current when they vibrate from the returning echo. After all echo data is received, the beamforming process (the compute intensive stage) combines the data into a partial image. The partial image corresponds to echoes from a single transmission. Several transmissions from different locations on the transducer array are needed to produce high quality images, so several iterations of transmit, receive, and beamsum are necessary to construct a complete frame.

Each transmission is a pulsed signal conceptually originating from a single location in the array, shown in 2.1a. Because the pulse is unfocused, it expands into the medium radially, and as it encounters interfaces between materials of differing density,

the signal will partially transmit and partially reflect as shown in 2.1b. The returning echoes cause the transducers to vibrate, generating a current signal that is digitized and stored in a memory array associated with each transducer. Each position within these arrays corresponds to a different round-trip time from the emitting transducer to the receiving transducer. Because transducers cannot distinguish the direction of an incoming echo, each array element contains the superimposed echoes from all locations in the imaging volume with equal round-trip times (i.e., an arc in the imaging volume). The delay and sum beamsum operation adds the echo intensity observed by all transducers for the arcs intersecting a particular *focal point* (i.e., a location in the imaging volume), yielding a strong signal (i.e., a bright point in the image) when the focal point lies on an echoic boundary.

The imaging volume geometry is described by a grid of *scanlines* that radiate at a constant angular increment from the center of the transducer array into the image volume. Focal points are located at even spacing along each scanline. In essence, the beamsum operations entail calculating the round-trip delay between the emitting transducer and all receiving transducers through a particular focal point, converting these delays into indices in each transducers' received signal array, retrieving the corresponding data, and summing these values. 2.1c illustrates this process. An image is formed by iterating over all desired focal points and performing beamsum for each. Once an image has been formed, a demodulation step removes the ultrasound carrier signal.

2.1.2 Delay Calculation

The delay calculation (identifying the right index within each receive array) is the most computationally intensive aspect of synthetic aperture beamforming as it must be completed for every {focal point, transmit transducer, receive transducer} trio. Typically, delays are calculated via

$$d_P = \frac{1}{c}\left(R_P + \sqrt{R_P^2 + x_i^2 - 2x_i R_P sin\theta}\right) \tag{2.1}$$

where d_P is the round-trip delay from the center transducer to the point P to transducer i, c is the speed of sound in tissue (1540 m/s), R_P is the radial distance of point P from the center of the transducer, θ is the angular distance of point P from the line normal to the center transducer, and x_i is the distance of transducer i from the center. 2.1d shows variables as they correspond to the system geometry. This formula applies the law of cosines to calculate the round-trip distance, and requires extensive evaluation of both trigonometric functions and square roots. Hence, many 2D ultrasound systems pre-calculate all delays and store them in a look-up table (LUT) [4, 43]. However a typical 3D system requires roughly 250 billion delay values, making a LUT implementation impractical. Instead, delays are calculated on-the-fly [98].

2.1.3 Receive Sub-aperture Multiplexing

Another challenge of 3D beamformation lies in managing the deluge of data that must be transferred from receiving transducers to functional units that perform the summation. In existing 3D ultrasound systems, the receive data is transferred from the scan head (which typically does not contain significant compute capability) via cable to separate systems that perform beamsum. For the transducer array geometry we assume, the receive data arrives at a rate of approximately 6Tb/s and comprises roughly 100MB per transmit; hence it is both too large to store in the scan head and arriving to quickly to transfer over cables.

To manage data transfer, modern systems employ *sub-aperture multiplexing*, wherein only data from a sub-aperture (a part of the imaging volume) are stored and transferred upon each transmit [37, 44]. Sub-apertures are sized based on the bandwidth available in the link from the transducer head to the computing platform. Transmission from a single source is repeated several times in succession, capturing a different

sub-aperture each time. Hence, over a sequence of transmits, all {transmit, receive, focal point} trios are obtained. Though effective at reducing data rates, this technique entails some compromise in image quality as neither the patient nor the scan head is entirely still between transmits, possibly resulting in some motion blur.

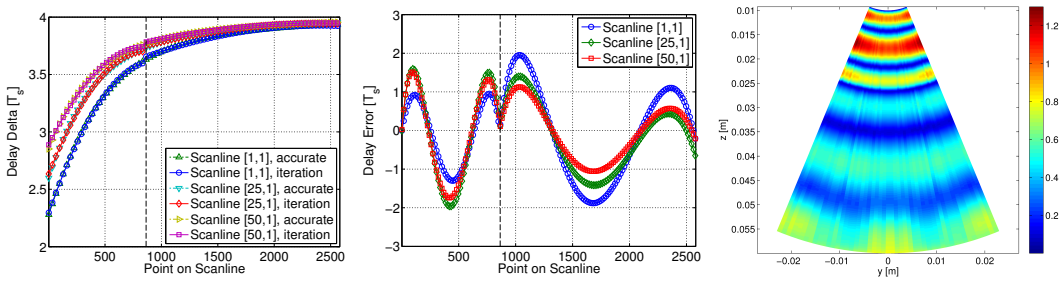
2.1.4 Virtual Source

In the simplified overview of ultrasound above, only a single transducer fires during each transmit operation. In fact, a single transducer produces too weak a signal to achieve high signal-to-noise ratio (SNR). To increase transmission power, modern ultrasound systems employ a *virtual source*, wherein multiple transducers together emulate a single point source located behind the transducer array [65]. The transducer array emulates the virtual source by timing and weighting the activation of several transducers such that they fire just as a "virtual" wave front from the virtual source passes through them. Beamformation proceeds as above, however, round-trip delays are calculated with respect to the virtual source position.

Activating more transducers increases signal strength, but can lead to interference effects and artifacts in the image. We have tuned our virtual source scheme in simulation to determine that transmission from a circle with diameter of 10 transducers (about 80 transducers in total) produces sufficient signal strength without interference or artifacts. Our findings match prior work [45].

2.2 Algorithm Design

In this section we describe the algorithmic innovation that was developed for Sonic Millip3De that allows the hardware to efficiently compute all of the delays without complex hardware or GB size look-up tables.



(a) **(b)** **(c)**

Figure 2.2: **Delay curve fitting and analysis.** (a) The exact delta between neighboring points for representative scanlines and the estimates from our iterative algorithm. The dotted line indicates the boundary of the 2-section piecewise approximation. (b) The error between our approximation and the exact delta, normalized to the index unit (T_s). With two sections, our algorithm never errs by more than 3 samples. (c) Root mean square error for an entire y-z image slice.

2.2.1 Iterative Index Calculation

As discussed previously, delay calculation is enormously compute-intensive, requiring either numerous processors or a large LUT of pre-calculated values. Neither of these approaches is feasible in a small hand-held device. Prior work has reduced delay calculation computational complexity through iterative methods [5, 54], but these methods still require billions of expensive square root operations. Instead, Sonic Millip3De uses a new algorithm to require tractable storage and eliminate the trigonometric and square root operations required in a straight-forward implementation.

The key insight of this algorithm is to replace prior iterative index calculations [54] with a piece-wise quadratic approximation that can be computed using only add operations. Because focal points are evenly spaced, the delta function between adjacent focal point delays form a smooth curve and indices can be approximated accurately (with error similar to that introduced by interpolation) over short intervals with quadratic approximations. These exact delta curves are replaced with a per-transducer pre-computed piece-wise quadratic approximation constrained to allow an index error of at most 3 (corresponding to at most 30μm error between the estimated

and exact focal point) thus resulting in negligible blur. Figure 2.2a compares the approximation to the exact difference between adjacent delays for three representative scanlines. Figure 2.2b shows the corresponding round-trip delay error. Figure 2.2c shows the root mean square (RMS) error for the full y-z slice through the middle of the image. This new approach drastically reduces storage requirements relative to pre-computing all delays because only few constants are pre-computed and stored per section. Because of its simplicity, this approximation requires only small table look-ups (to retrieve constants) and adds (to iteratively calculate the delay).

2.2.2 Narrow-Width Fixed-Point Arithmetic

Sonic Millip3De also employs narrow-bit-width fixed-point arithmetic to further reduce the storage and bandwidth requirements of our design. We performed an offline study to analyze image quality at various bit widths. The analysis of this study (see Section 2.4.2) concludes that 12-bit fixed-point precision sacrifices negligible image quality relative to double-precision floating point, but any further bit-width reduction leads to significant quality degradation. Using custom-width 12-bit ADCs and calculation pipelines substantially reduces hardware and power requirements relative to conventional 16-bit or wider DSP solutions [4, 43].

2.3 Hardware Architecture

We next describe the original Sonic Millip3De system architecture [74, 76] and its key features, including the beamforming accelerator that implements the previously described iterative delay calculation algorithm in a massively parallel array.

2.3.1 System Architecture Overview

The Sonic Millip3De system (Figure 2.3) comprises three stacked silicon dies (transducers and analog electronics, ADC and storage, and computation) connected

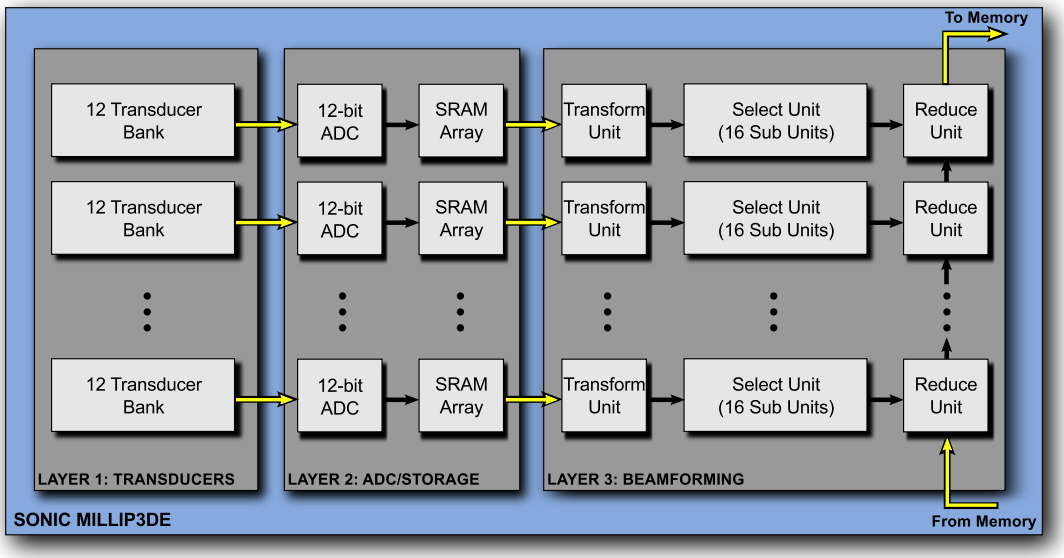


Figure 2.3: **Sonic Millip3De Hardware Overview.** Layer 1 (24×18mm) comprises 128x96 transducers grouped into banks of 3x4 transducers each. Analog transducer outputs from each bank are multiplexed and routed over TSVs to Layer 2, comprising 1024 12-bit ADC units operating at 40MHz and SRAMs arrays to store incoming samples. The stored data is passed via face-to-face links to Layer 3 for processing in the 3 stages of the 1024-unit beamsum accelerator. The interpolation stage upsamples the signal to 160MHz. The 16 units in select stage map signal data from the receive time domain to the image space domain in parallel for 16 scanlines. The summing stage combines previously-stored data from memory with the incoming signal from all 1024 beamsum nodes over a unidirectional pipelined interconnect, and the resulting updated image is written back to memory.

vertically using through-silicon vias (TSVs) and off-stack LPDDR2 memory. These components are integrated in the ultrasound scanhead, the wand-like device a radiologist manipulates to obtain ultrasound images. Our design focuses on the so-called "front-end" of an ultrasound system, which controls the transducer array and constructs a volumetric image. A separate "back-end" renders a view (either 2D slices or a 3D perspective) of the image; however, the design of a "back-end" system such as a tablet or other presentation system that can interface with the scanhead is not considered as part of this work.

We split the design over three 3D-stacked layers for several reasons. First, the technology requirements of each layer differ substantially. The geometry of the trans-

ducer array requires a much larger die and higher voltages than the SRAM arrays or beamforming accelerator and can be economically manufactured in an older process technology. In contrast, the power hungry ADC/memory and computation layers benefit from exploiting the latest process technology.

Second, the layout of the transducer array is tightly coupled to the transmit frequency and target imaging aperture; ultrasound systems typically feature interchangeable scan heads with varying array geometries for different imaging tasks (e.g., different imaging depths). By separating the transducer array, ADC/storage, and computation engine into separate dies, a standard interface (i.e., TSV layout) between each enables dies to be reused with varying transducer array layers, reducing design costs.

Finally, as in recent 3D-stacked processor architectures where caches and cores are connected vertically [28], the face-to-face connections between SRAM arrays and corresponding computation units avoid the need for long wires.

For this original design, the transducer die comprises a 128×96 grid of optimally spaced transducers whose centers are exactly $\lambda/2$ apart where λ is the wavelength of the transmit signal [52]. We assume a 4MHz transmit frequency, requiring a minimum die size of 24mm×18mm—much larger than the other layers, which are 15mm×15mm each. The area between transducers contains the analog electronics and routing to the TSV interface to the ADC/storage die. Transducers are grouped into 1024 banks of twelve (3×4) transducers each. One transducer within each bank is assigned to one of twelve receive sub-apertures. During each transmit cycle, only a single transducer among the twelve in each bank will receive data and pass it to the ADC layer. Hence, twelve consecutive transmits are required to process the entire aperture. Transducers within a bank are multiplexed onto a single signal per bank that is passed over a TSV to the ADC/storage layer for digitization.

The ADC/storage layer comprises 1024 12-bit ADCs, each connected to an in-

coming analog signal from the transducer layer. The ADCs sample at a frequency of 40MHz, well above the Nyquist limit of even the fastest transducer arrays (15MHz). This sampling frequency balances energy efficiency and flexibility for ultrasound applications requiring varying transmit frequencies. After digitization, the received signals are stored in 1024 independent SRAM arrays, each storing 4096 12-bit samples. The SRAMs are clocked at 1GHz. Each SRAM array is connected vertically to a corresponding functional unit on the computation layer, requiring a total of 24,000 face-to-face bonded data and address signals.

The computation layer is the most complex of the three. It includes the beamforming accelerator units, a unidirectional pipelined interconnect, a control processor (e.g., an M-class ARM core), and an LPDDR2 memory controller. The die area is dominated by the beamforming accelerator array and interconnect, which are described in the following subsections. The control processor manages memory transfers from the LPDDR2 interface to the accelerator array, controls the transducer array, and performs other general purpose functions (e.g., configuration, boot). The off-stack LPDDR2 memory stores index delay constants and a frame buffer for the final volumetric image. While the control processor has a small cache, the accelerator array performs only bulk memory transfers and requires no cache hierarchy or coherence mechanism.

The Sonic Millip3De memory system comprises a 192-bit wide memory channel striped across 6 2Gb x16 LPDDR2-800 parts. This unusual arrangement matches the width of our on-chip interconnect, provides sufficient capacity (1.5 GB) and sufficient memory bandwidth (38.4 GB/sec) to load beamforming constants (requiring 6.2 GB/sec) and read/write image data (requiring 5.5 GB/sec) for our target imaging rate of one frame per second while still requiring little power (see 2.4) [55, 64].

2.3.2 The Beamforming Accelerator

The beamforming accelerator is the central element of Sonic Millip3De, and is the key to achieving our performance and power objectives. The accelerator relies on massive parallelism (1024 beamforming units operate in concert) and achieves energy efficiency through carefully optimized 12-bit data paths that perform only add, compare, and table look-up operations.

As described before, a single ultrasound frame is obtained by summing the received data from 12 receive sub-apertures over 16 different virtual sources. For each of these 192 receive operations, the entire imaging volume is read from memory (15MB), the (single) correct sample from each transducer in the sub-aperture is added to each focal point, and the volume is stored back to DRAM. Below, we describe a single of these 192 receive operations: the data flow during each receive is identical, only the apodization and delay constants differ across receives.

2.3.2.1 Beamforming Node.

Each beamforming node comprises an interpolation unit, 16 select sub units, and a single summation unit linked to the global interconnect. There are 1024 such nodes, each connected to a corresponding ADC/SRAM channel and transducer bank. Each node occupies roughly 400μm $\times$ 400μm (see 2.4).

2.3.2.2 Interpolation Unit.

The interpolation unit includes a linear interpolation unit, which upsamples the transducer data upon request from a select unit. Upsampling is a standard technique in ultrasound to improve image resolution without the power overhead of faster ADCs [14]. This unit also applies an apodization constant which corresponds to the weighting of the channel. Channels near the edges of the array are given less weight because they are more likely to produce artifacts from side lobes.

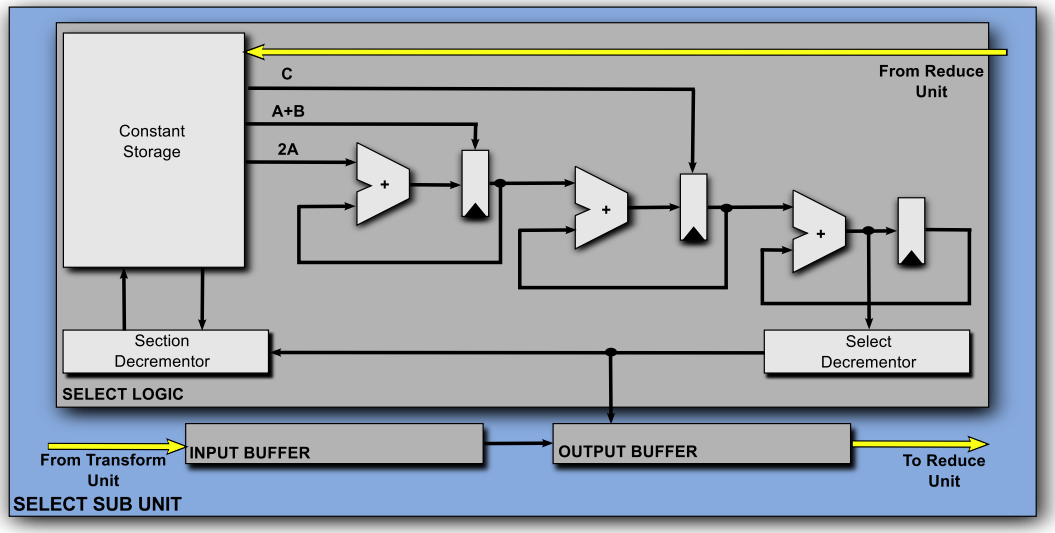


Figure 2.4: **Select Unit Microarchitecture.** Select units map incoming samples from the receive time domain to image focal points. Sample data arrives from the interpolation unit at the input buffer, and each sample is either discarded or copied to the output buffer to accumulate a particular focal point. The unit selects the correct sample for each focal point using the indexing algorithm in Section 2.2. The Constant Storage holds the 3 approximation constants and boundary for each approximation section. The first adder calculates $2AN + A + B$, the second adds the $N - 1$ result of the quadratic equation to create the value for N, and the final adder accumulates fractional bits from previous additions. The Select Decrementor is initialized with the integer component of the sum. Each cycle, the head of the input buffer is copied to the output if the decrementor is zero, or discarded if it is non-zero. The Section Decrementor tracks when to advance to the next piece-wise approximation section.

2.3.2.3 Select Units.

Select units map the interpolated receive data to focal points for a single scanline using the algorithm described in 2.2. Because the unit processes focal points in order of increasing distance from the scanhead, the round-trip delays increase monotonically (2.2a). Hence, the unit can select the correct sample for each focal point in a single pass over the receive data.

The select unit block diagram is shown in 2.4. The *Constant Storage* block stores the delay constants and section boundaries used in the index approximation algorithm. Constants are loaded between each scanline bundle. The *Input Buffer* and *Output Buffer* are FIFO queues. Also shown are three adders, which calculate the

next delay delta via our quadratic estimation, and two decrementors, which orchestrate input data selection and the piece-wise quadratic sectioning.

Whenever the Input Buffer is empty, the sampling unit requests the next 16 12-bit samples from the interpolation unit. The select unit then generates the index of the first focal point on the scanline by adding the transmit (Tx) and receive (Rx) constants, placing the sum in the *Select Decrementor*. Each cycle, the Select Decrementor decrements. If the value is non-zero, then the head of the input buffer is discarded—that input sample does not correspond to the next focal point. However, when the value becomes zero, the head of the input buffer is appended to the output buffer—this input sample will be added to the next focal point.

The Section Decrementor counts down the remaining focal points in the current section. When it reaches zero, the constants and boundary for the next approximation section are loaded from the Constant Storage.

Through this simple use of decrementors, adders, and a few pre-computed constants, the sampling unit completely avoids the need for complex delay calculations. It is this simple design that enables the enormous energy efficiency gains of the Sonic Millip3De.

2.3.2.4 Summing Unit & Interconnect.

The summing unit contains an array of 16 192-bit buffers (one per scanline) and 16 12-bit adders. Whenever both the buffer and select unit for a particular scanline are ready, the values are added and passed to the next beamforming node.

The summing units are connected via the unidirectional pipeline interconnect. Each link is 192 bits wide, and clocked at 1GHz. The network provides a peak bandwidth of 22.3GB/s between neighboring nodes, comfortably exceeding the minimum requirements to achieve 1 frame per second. The links between beamforming units are nearly 400μm long, and are routed on a quad-spaced metal layer. The wires

are not repeated, as an entire clock cycle is available to traverse between units. Because of the sheer number of wires (192 links each between 1024 beamforming units), the interconnect accounts for a substantial fraction of the overall power of the Sonic Millip3De system (see 2.4).

2.3.2.5 Processing Data.

Once a receive operation is complete (all SRAM buffers are filled) the control unit activates the accelerator. The entire beamforming array processes only 16 adjacent scanlines at a time, traversing the entire input data in the SRAMs for these scanlines before reprocessing the data for the next 16-scanline bundle; 157 such bundles are processed.

Between bundles, receive delay and apodization constants are loaded from memory and fed into all the beamforming units by sending control packets addressed to each unit around the interconnect. About 250kB of constants must be loaded per bundle.

The select units within each beamforming node all operate independently—each selecting focal point data for one of the 16 scanlines. The select units arbitrate for access to the interpolation unit and request the next 4 transducer samples, which the interpolation unit interpolates and apodizes to produce 16 properly weighted samples. Select units continue requesting input data and outputting focal point data until they fill their 16-entry output buffer, at which point they block. The select unit microarchitecture is detailed in 2.4.

In the mean time, image data from earlier receives is loaded into the network's ingress node. Data is read from each of the 16 active scanlines in a round-robin fashion, 16 focal points (192 bits) at a time, and injected into the network.

For a particular scanline, when both the select unit's output and the data arriving via the interconnect are available, the two are added and propagated to the next beamforming node. Thus, each 16-point bundle from each scanline will visit every

Parameter	Value
Sub-apertures	12
Virtual Sources	16
Total Transmits per Image	192
Total Transducers	12,288
Receive Transducers per Sub-aperture	1024
Storage per Receive Transducer	4096 x 12-bits
Focal Points per Scanline	4096
Image Depth	6cm
Image Total Angular Width	$\pi/2$
Sampling Frequency	40MHz
Interpolation Factor	4x
Interpolated Sampling Frequency	160MHz
Speed of Sound (tissue)	1540m/s
Target Frame Rate	1fps

Table 2.1: 3D ultrasound system parameters.

beamforming node, accumulating the appropriate incoming sample. The scanline flows circle the network independently. When the last data return to the control processor at the egress node, the next bundle of scanlines is processed. When all bundles have been processed, the SRAM buffers on the ADC/storage layer are cleared and the control processor triggers the next transmit.

2.4 Full-System Evaluation via Simulation

In this section we cover simulated results and evaluation of the entire Sonic Millip3De design. First, we validate that algorithmic approximations and fixed-point rounding errors do not compromise image quality. Second, we report full-system power requirements in 45nm, and project when technology scaling will enable a 5W objective for safe human skin contact.

2.4.1 Evaluation Methodology

To evaluate our full-system design, we use the ultrasound parameters shown in Table 2.4. We analyze image quality using Field II [41, 42] (a widely-used simulation framework for ultrasound imaging built on top of MATLAB) to simulate echo signals

Bits	10	11	12	13	14	Ideal
CNR	2.233	2.536	2.942	2.960	2.942	2.972

Table 2.2: **CNR vs. precision.** Ideal indicates double precision floating point and exact delay calculations.

of a 5mm cyst in tissue and visualize the resulting images using MATLAB.

The objective of this evaluation is to measure the image quality loss that arises due to Sonic Millip3De's iterative delay calculation and 12-bit fixed point functional units. Hence, we contrast this new algorithm against a double-precision floating point simulation using precise delays (calculated via Eq. 5.1).

We quantify image quality by measuring contrast-to-noise ratios (CNR) of imaging the 5mm cyst. CNR is a standard measure of the accuracy of ultrasound imaging [17, 84]; it measures the contrast resolution of echoic regions (tissue) and anechoic regions (cyst) under the effects of receiver noise and clutter (reflections from outside the imaging aperture that produce artifacts within the image). CNR is defined as:

$$CNR = \frac{|\langle L_{cyst} \rangle - \langle L_{background} \rangle|}{\sqrt{\sigma^2_{cyst} + \sigma^2_{background}}} \tag{2.2}$$

where $\langle L_{cyst} \rangle$ and $\langle L_{background} \rangle$ are the mean signal within the cyst and background areas, respectively, and σ^2_{cyst} and $\sigma^2_{background}$ are the corresponding variances.

To measure hardware power and performance, we synthesize an RTL-level specification in Verilog using an industrial 45nm standard cell library and SRAM compiler. We report power, timing, and area estimates from synthesis. We model interconnect power and performance using SPICE. We estimate ADC power from recently published designs in 40nm [59, 87]. The cited ADC designs provide 11-bit precision and operate at 6× higher frequency than our system require. We scale down operating frequency by 6×, but conservatively estimate that increasing precision to 12 bits quadruples power requirements, for a net power scaling factor of 2/3. We estimate DRAM power from reported efficiency of LPDDR2 designs [55], assuming 12 x16 2Gb

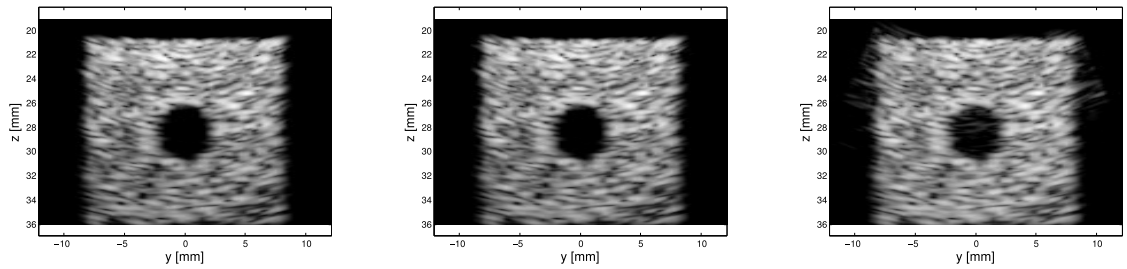


Figure 2.5: **Image Quality Comparison.** (a) Y-Z (vertical) slice through cyst from a 3D simulation using Field II [41, 42], generated with double-precision floating point and exact delay calculation (Eq. 5.1). CNR is 2.972. (b) The same slice generated via our delay algorithm and 12-bit fixed-point precision. CNR is 2.942. (c) Same as (b), with 11-bit precision. CNR is 2.536.

parts.

2.4.2 Full-System Image Quality

We contrast images of our cyst model using precise index calculations (i.e., Equation 5.1) and double-precision floating-point against the same image reconstructed via our iterative delay method and 12-bit fixed-point precision as implemented in the Sonic Millip3De hardware. 2.5 (a) shows a slice from the baseline double-precision simulation, while (b) shows the same slice using our methods and fixed-point beamsum. The baseline algorithm achieves a CNR of 2.972, while our design produces a nearly indistinguishable image with CNR of 2.942 (higher values indicate better contrast). Reducing precision to 11 bits (c), however, results in noticeable artifacts and CNR of only 2.536. 2.2 shows CNR for a range of precision.

2.4.3 Full-System Power

We next report Sonic Millip3De's power requirements in 45nm technology and project requirements in future nodes; scaling trends are shown in 2.6 while a detailed breakdown appears in 2.3. Using a combinations of synthesis results and quoted estimates ([55, 87]), we determine that the Sonic Millip3De system requires a total of

	Transducers	ADC [87]	SRAM	Accelerator	DRAM [55]	Total
45nm	0.3W	1.2W	0.197W	9.84W	3.8W	15.3W
11nm	0.3W	0.146W	0.049W	2.43W	0.461W	3.39W

Table 2.3: **Power Breakdown.** SRAM power from an industrial 45nm SRAM compiler. Accelerator power from synthesized RTL using 45nm standard cells for logic and from 45nm SPICE simulations for interconnect. Scaling prediction trends from [23, 59].

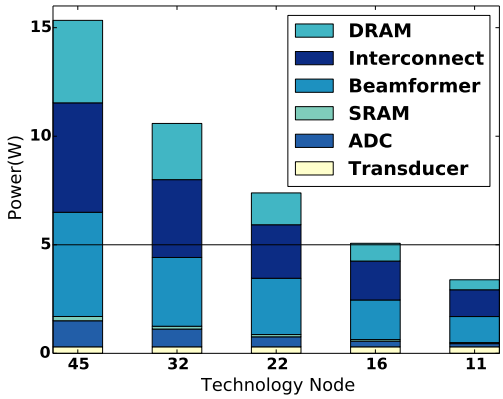


Figure 2.6: **Power Breakdown Across Technology Nodes.** Scaling projections based on trends reported in [23, 59]. We project meeting the 5W power budget at the 11nm node.

15.3W in 45nm, falling short of the target 5W power budget for safe use on humans. However, we note that over 60% of the power is dissipated in the compute layer. Hence, further architectural and circuit innovation and technology scaling can close the power gap.

Finally we project when Sonic Millip3De might meet the power target using published scaling trends. For this analysis, we use ADC scaling trends from [59], technology scaling from [23], and assume that wire power does not scale (though our wires get shorter from transistor shrinking). We project that Sonic Millip3De will fall just short of our goal in 16nm, and meet the 5W target by the 11nm node.

2.5 Accelerator Evaluation via FPGA

In this section we present image quality and performance results from an FPGA implementation of a single channel of the accelerator layer [77]. While this evaluation only covers a single channel, it serves as a proof-of-concept for the full design.

2.5.1 Data Collection

2D: We acquire RF echo data from an artificial tissue phantom using a Philips P4-1 operating at a 2.5 MHz center frequency. The transmission pulses of all 96 transducers are time-delayed to create a virtual source located 1 mm behind the transducer head. We apply Hamming window apodization to the transmit pulses. We collect the unprocessed RF data with a Verasonics V-1 system and then perform beamforming on the FPGA.

3D: Currently, commercially available 3D ultrasound probes do not provide access to unprocessed RF data. Hence, we use synthetic echo data, simulated in Field II [41, 42], to create an input data set for 3D beamforming. We simulate a 32x32 transducer receive aperture and transmit from a center circle of 78 transducers. Similar to the 2D methodology, transmission pulses are time-delayed to create a virtual source 1 mm behind the array.

2.5.2 Beamforming

We evaluate image quality by contrasting three beamforming methods for both 2D and 3D echo data using the parameters described in Table 2.4. First, we generate a ideal baseline image using MATLAB and double precision floating-point, calculating delays precisely based on Euclidean distance. Second, we model the operation of the hardware in MATLAB, using 12-bit fixed-point arithmetic and the iterative delay calculation approach, as described in 2.2. Finally, we perform beamforming in hardware on the DE1 FPGA board.

Parameter	Value
Sub-apertures	12
Virtual Sources	16
Total Transmits per Image	192
Total Transducers	12,288
Receive Transducers per Sub-aperture	1024
Storage per Receive Transducer	4096 x 12-bits
Focal Points per Scanline	4096
Image Depth	6cm
Image Total Angular Width	$\pi/2$
Sampling Frequency	40MHz
Interpolation Factor	4x
Interpolated Sampling Frequency	160MHz
Speed of Sound (tissue)	1540m/s
Target Frame Rate	1fps

Table 2.4: 3D ultrasound system parameters.

In all scenarios, we pre-process RF data using time-gain compensation. We truncate input signals to 12-bit precision for the fixed-point methods. After beamforming, we perform demodulation and Cartesian regridding (to facilitate easier viewing of 3D images) in software.

2.5.3 FPGA Processing

We implement a Sonic Millip3De beamforming channel on an Altera DE1 board which features a Cyclone II FPGA chip. This FPGA is quite small, and can accommodate only a single beamforming channel with a half-width select unit (processing eight rather than 16 scanlines concurrently).

We iteratively load each transducer's receive signal into the FPGA and process the channel. To automate channel processing, we implemented a MATLAB program that manages communication with the FPGA, iteratively loading the beamforming coefficients and RF data. The FPGA then fully processes the image volume for the single channel, producing eight scanlines at a time in repeated passes over the channel data. The partial sums from all channel are then accumulated in MATLAB.

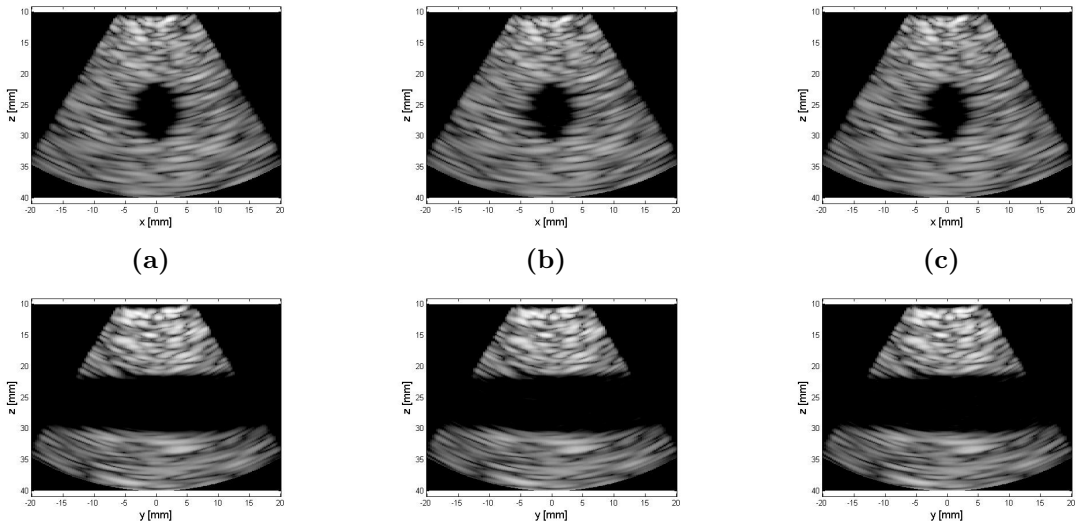


Figure 2.7: **3D Image Quality Comparison.** Beamformed images of a simulated 3D cylindrical cyst using Field II [41, 42]. (a-c) depict x-z slices and (d-f) depict y-z slices. (a,d) Generated in MATLAB with full delay and double-precision floating-point (b,e) Generated in MATLAB with 12-bit fixed-point and iterative delay estimation (c,f) Generated on the FPGA prototype.

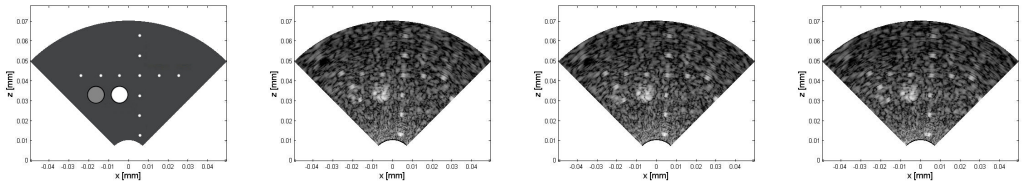


Figure 2.8: **2D Image Quality Comparison.** (a) Phantom guide. Phantom features two hyperechoic gray scale targets at 3 cm: +6 dB (left) and >+15 dB (right). (b) Image generated with double-precision floating point and exact delay calculation. (c) The same image generated via our delay estimation and 12-bit fixed-point precision in MATLAB. (d) Image generated on the FPGA prototype.

2.5.4 FPGA Image Quality

Fig. 2.7 shows x-z and y-z slices of the 3D results and Fig. 2.8 shows the beamformed image results from the 2D phantom data. By visual comparison, the images are all similar with notable objects equally identifiable in the images with no apparent degradation or artifacts.

To compare the images quantitatively, we report the root-mean-square difference

Table 2.5: Root-mean-square difference in pixel value across each beamforming method pair, expressed as a percent of the 40 dB dynamic range.

Image Pair	2D RMS	3D RMS
Float-Fixed	6.8%	0.7%
Float-FPGA	9.9%	0.5%
Fixed-FPGA	9.4%	0.6%

in pixel values, expressed as a percent of the 40 dB range, for each pair of beamforming methods (shown in Table 2.5). (These reflect the entire image volume, not just the shown slices). The small differences, particularly for 2D, are primarily a consequence of rounding errors in type conversion when pre-processing the RF data to load it onto the FPGA.

2.5.5 FPGA Performance

We report performance (time to compute a frame) of the FPGA and contrast it with the projected performance of a full-scale Sonic Millip3De ASIC implementation. The FPGA prototype runs at 50 MHz and computes a partial image in 0.2 seconds per transducer channel in 2D, and 2.3 seconds per transducer in 3D. When scaled based on our prior estimates 2.4 for 1 GHz ASIC clock frequency (20x speedup), 16 instead of 8 way parallelism in select units (2x speedup), and individual high-speed memories (estimated 24x speedup), we project that a full-scale Sonic Millip3De implementation can generate 2D images at over 4k frames per second and 3D images at over 400 volumes per second for the system configurations described in Table 2.4.

Finally, the 3D results in Fig. 2.7 are much shallower than the results we simulated in 2.4 and are generated from a single firing of one sub-aperture. This was required as generating images from the full-aperture and compounding with the single-channel FPGA would have not been feasible. Therefore to study the runtime of such a system, we scale up the recorded runtime from 2.3 seconds to 5 seconds to account for a 10 cm deep image and require the 32x32 aperture to be processed 192 times to complete

the full compounding, resulting in 960 seconds to complete the image based on the FPGA runtime. Scaling this time to the ASIC clock speed (20x), doubling the select sub-units to 16 (2x), and accounting for the ASIC memory system (24x), we find that the full-system would achieve 1 frame per second which matches the estimated results from simulation.

2.6 Conclusions

In this chapter, we have described Sonic Millip3De, a new 3D stacked accelerator unit for hand-held 3D ultrasound. This design combines a streaming accelerator architecture with a newly developed iterative delay calculation algorithm to minimize power requirements and exploit data locality. Using synthesis of RTL-level hardware design for an industrial 45nm standard cell process, we have shown that this design can enable volumetric imaging with a fully sampled (128×96) transducer array within a 16W full-system power budget. Based on current scaling trends, we project that the Sonic Millip3De will meet the 5W target power budget for safe use on humans by the 11nm technology node. Additionally, we have shown a proof-of-concept, scaled-down FPGA implementation to further validate the design.

CHAPTER III

High Frame Rate Sonic Millip3De

In this chapter we cover our follow-up work to the Sonic Millip3De design to substantially increase frame rate using separable beamforming and new firing schemes with the contents of this chapter primarily taken from [75, 93, 94, 95, 96]. These algorithmic improvements, developed with collaborators at Arizona State University, expand the capabilities of our redesigned Sonic Millip3De to support faster real-time imaging as well as enable the hardware to support a wider range of beamforming methods.

We begin by introducing a new firing scheme that provides a 2× speedup without any changes to the hardware, increasing the frame rate of the original design. We then discuss a revised hardware design that leverages a 2-stage separable beamforming algorithm and new firing scheme to achieve up to 32× volumes per second with no significant image quality degradation. Finally we illustrate the hardware performance with planar wave transmit which enables over 1k volumes per second to produce B-mode images suitable for subsequent image analysis tasks, such as flow imaging.

3.1 Sliding Sub-aperture

Sub-aperture-based processing [37, 44] is one way of reducing the number of concurrent active channels, thereby reducing the computation load per firing. Recall

that in the original Sonic Millip3De design, previously outlined in Chapter II, we used a sub-aperture technique wherein the 128×96 element array was divided into 12 non-overlapping sub-apertures. For each firing (of the 16 virtual sources), one of the sub-apertures would receive data, and the process was repeated until every virtual source/sub-aperture pair was completed, giving 192 total firings per frame.

In order to reduce the firings per frame while still achieving comparable image quality, we use a new scheme we developed that keeps the same sub-aperture size (1,024 elements to match the 1,024 channel hardware) but instead uses overlapping sub-apertures. This new subaperture firing scheme used is illustrated in Fig. 3.1a. 120×88 array elements are organized into 96 overlapping sub-apertures of 32×32 elements each. All 32×32 elements of each sub-aperture are used for receive, while for transmit, only the center 76 array elements of the sub-aperture are activated as a virtual source. 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. Using this method reduces the total firings per frame from 192 to 96, giving over a 2×speedup in frame rate by only modifying the transducer layer to 120×88 from the original 128×96.

3.1.1 Apodization for Overlapping Sub-apertures

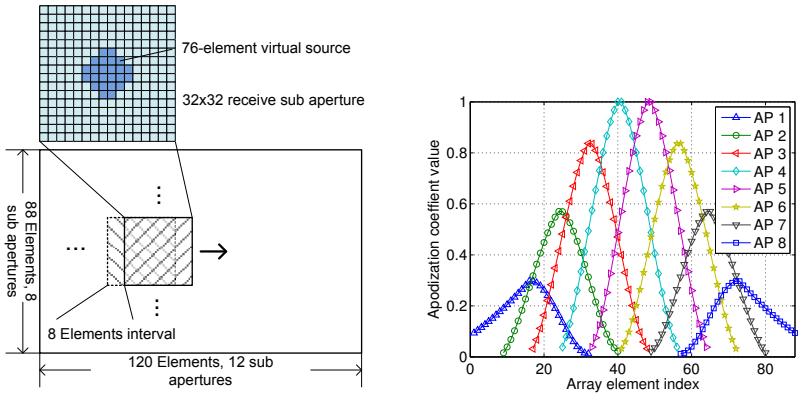
As discussed in the previous chapter, a constant weighting factor, or apodization, is applied to each channel based on the channel's traducer position. This limits the contribution of transducers near the edge, which typically view the image space from a wider angle, reducing sidelobe artifacts. Due to the overlapping design, each transducer can receive signal on multiple sub-apertures and in different positions relative to each sub-aperture it is a part of. Therefore, the weighting that is applied is no longer static for each channel and instead must be optimally calculated.

The results in this chapter use a method from [95] which computes each apodization value by combining a localized apodization window based on transducer position within the sub-aperture with a global window based on the overall position. This information is also combined with how many sub-apertures a given transducer is part of. Due to the sliding overlap of sub-apertures, ones near the edge of the array don't contribute to as many sub-apertures. Fig 3.1b shows the optimized windows used for the y component. A similar scheme is used in x, and the two are multiplied to give the final values. As the image shows, each sub-aperture has a localized window applied. However, the windows are scaled and corrected such that the summation of all of the local windows results in the correct global window function on the entire aperture. The correction is most noticeable near the edges where the number of sub-apertures each transducer contributes to varies the most. More detail of the computation of these windows can be found in [95].

However, for the benchmark suite discussed in Chapter V, a simpler apodization method is used. Instead of calculating an optimal solution that is specific to a given aperture/sub-aperture scheme, a more generic, two-stage Hamming window is used. In this method a Hamming windows, given by

$$w(n) = 0.54 - 0.46\cos(\frac{2\pi n}{N-1}) \tag{3.1}$$

where N is the window length and n is the element, are used. In this two-stage method, each element has a local Hamming window applied based on its position in the sub-aperture and the size of the sub-aperture as well as a global Hamming window applied based on size and position in the full aperture. As before, this technique is done for both x and y positions in the array, with the final value per element being a product of four apodizations (two local and two global). Overall this method is much more general and can be applied to any sub-aperture scheme, and while it is not as



(a) Improved Firing Method **(b)** Modified Apodization

Figure 3.1: **Sliding Sub-aperture:** (a) Faster sub-aperture firing technique that trades the 12 non-overlapping 32×32 sub-apertures of the original design for a sliding 32x32 subaperture that is shifted by 8. Additionally, virtual sources are located in the center of the sub-aperture in this scheme, as opposed to 16 virtual sources that are fired for all sub-apertures. (b) y component of apodization values of each element. Because the elements are used in varying number of sub-apertures, apodization must be modified to create correct window over entire receive aperture.

optimal as the method above, our simulations show that CNR reduction is only a few percent versus the optimally cacluated method for the sub-aperture parameters we use.

3.2 Separable Beamforming

While it does not require any modifications to the hardware, the sliding sub-aperture benefit is fairly limited. To improve the frame rate achievable with Sonic Millip3De, we redesigned the accelerator to leverage a new algorithm for beamforming that separates beamforming into two simpler stages. The separable beamforming algorithm we use was first introduced in [93, 95] and in my collaborator Ming Yang's PhD dissertation [97].

Figure 3.2 illustrates the principle of separable beamforming, wherein the first stage performs beamforming along the x axis by steering on the azimuth angle, while the second stage uses the partially beamformed output of the first stage to steer

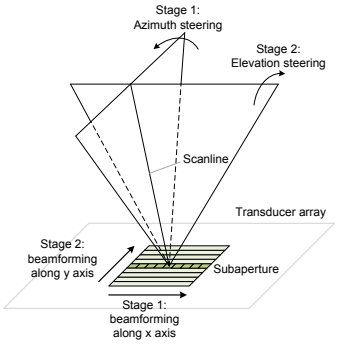


Figure 3.2: The principle of separable beamforming

along the elevational angle. Our contribution here is to adapt the Sonic Millip3De hardware design to facilitate this two-stage beamforming with two passes through the accelerator, yielding a substantial reduction in computation per volume and a significant increase in frame rate.

3.2.1 Background: Separable Beamforming

Table 3.1: 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	2
A/D sampling rate, MHz	40

We briefly summarize separable beamforming process, with reference to the 3-D ultrasound system described by the configuration shown in Table 3.1 and the 3-D coordinate system shown in Fig. 3.3. Let (R, θ, ϕ) be the coordinates of a focal point P. Here R is the radial distance from the origin O to point P. Point P' is the orthogonal projection of P in the yz plane. ϕ is the elevation angle between line OP' and the z axis. θ is the azimuth angle between OP and its orthogonal projection OP'

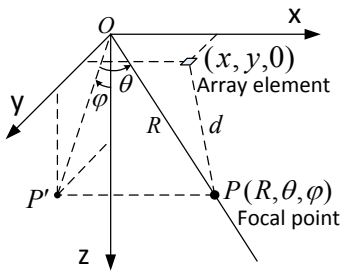


Figure 3.3: 3-D coordinate system used in the beamforming formulation

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

$$d_{\text{rx}} = \sqrt{R^2 + x^2 - 2Rx\sin(\theta) + y^2 - 2Ry\cos(\theta)\sin(\phi)} \tag{3.2}$$

Similarly the distance between the firing virtual source located at (x_v, y_v, z_v) and the focal point P is given by

$$d_{\text{tx}} = \sqrt{R^2 + x_v^2 + y_v^2 + z_v^2 - 2x_vR\sin\theta - 2Ry_v\cos\theta\sin\phi - 2Rz_v\cos\theta\cos\phi} \tag{3.3}$$

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

$$\tau(x, y, R, \theta, \phi) = (2R - d_{\text{tx}} - d_{\text{rx}})/c \tag{3.4}$$

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 conventional (non-separable) beamforming for subaperture l whose left corner indices are i_l and j_l, is given by:

$$F_l(m_R, m_\theta, m_\phi; t) = \sum_{n_x=i_l}^{i_l+N_x-1} \sum_{n_y=j_l}^{j_l+N_y-1} A_l(n_x, n_y) \cdot S_l(n_x, n_y, t - \tau(n_x, n_y, m_R, m_\theta, m_\phi)) \tag{3.5}$$

where $S_l(n_x, n_y, t)$ is the signal received by transducer element (n_x, n_y) at lth firing 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, which is sampled at $t = 2R/c$ for dynamic focusing. For a synthetic aperture ultrasound system, the final high resolution image is obtained by summing all the low resolution images from all subapertures.

In the separable beamforming we employ, $\tau(n_x, n_y, m_R, m_\theta, m_\phi)$ is instead approximated by a decomposition:

$$\tau(n_x, n_y, m_R, m_\theta, m_\phi) = \tau_1(n_x, n_y, m_R, m_\theta) + \tau_2(n_y, m_R, m_\theta, m_\phi) \tag{3.6}$$

wherein τ_1 drops cross-terms involving m_ϕ while, τ_2 drops cross-terms involving n_x.

The original beamforming problem (Eq. 3.5) can then be represented as a two-stage process:

$$F_l^{(1)}(n_y, m_R, m_\theta; t) = \sum_{n_x=i_l}^{i_l+N_x-1} A_l(n_x, n_y) S_l(n_x, n_y, t - \tau_1(n_x, n_y, m_R, m_\theta)) \tag{3.7}$$

$$F_l^{(2)}(m_R, m_\theta, m_\phi; t) = \sum_{n_y=j_l}^{j_l+N_y-1} F_l^{(1)}(n_y, m_R, m_\theta; t - \tau_2(n_y, m_R, m_\theta, m_\phi)) \tag{3.8}$$

In the first stage, the beamforming is along the x axis, which functions as a spatial filter that steers the receive plane to azimuth angle θ. The process repeats for all

combinations of m_R, n_y and m_θ and results in a partially beamformed intermediate signal $F_l^{(1)}$. In the second stage, 1-D beamforming is performed along the y axis, and corresponds to steering receive plane to elevation angle ϕ. The second stage beamforming is repeated for all combinations of m_R, m_θ and m_ϕ.

The separable formulation drastically reduces the total number of delay-sum operations. 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 conventional, non-separable beamforming. Thus, the computational complexity reduction is $N_x M_\phi/(N_x + M_\phi)$. For the configuration shown in Table 3.1 with a 32×32 subaperture size and 48×48 scanlines, this approach achieves about $19\times$ complexity reduction.

The accuracy of the separable method depends on the quality of the approximations $\tau_1(x, y, R, \theta)$ and $\tau_2(y, R, \theta, \phi)$. [95] develops a formulation of τ_1 and τ_2 that minimizes RMS phase error while ensuring the means of τ_1 and τ_2 are the same, which balances the required data buffering between the two beamforming stages.

3.2.2 Online Iterative Separable Delay Calculation

As with the non-separable beamforming described in Chapter II, our separable beamforming implementation approximates the delta between consecutive values of τ_1 and τ_2 to avoid impractically large look-up tables: for our system configuration, look-up tables of τ_1 and τ_2 for 96 subapertures include at least (considering symmetry) 5.7 billion and 8.9 billion constants, respectively.

We use piece-wise quadratic curves to approximate the delay difference between consecutive samples along a scanline. Instead of storing the delay look-up table directly, the coefficients a, b and c of the quadratic approximation and the initial delay are stored, and the delays are iteratively calculated using these coefficients. The iterative calculation method does not require multiplications, it can be implemented

in a simple circuit using only three additions.

To get an accurate approximation, each scanline is divided into 2-4 sections and the delay in each section is approximated by a quadratic curve. For our system configuration, where the depth ranges from 2cm to 10cm, we cannot use a 2 section configuration since it results in significantly large approximation error. We choose a 3 section configuration over a 4 section configuration since it requires 23% lower storage with comparable approximation error.

The storage requirements of this method are as follows. Each section is characterized by three constants and an initial point, and each scanline requires an additional start index. Thus, each scanline requires 13 constants. A total of 38M constants must be stored; 15M constants are required for τ_1 and the remaining 23M for τ_2. The 15M constants for τ_1 correspond to 13 constants/scanline $\times$ 48 scanlines $\times$ 1,024 transducers/subaperture $\times$ 96 subapertures, divided by 4 due to symmetry (the delay term is symmetric in both x dimension and y dimension and so it is sufficient to store only 1/4 of the constants). The number of constants for τ_2 is calculated in a similar way. Each constant requires 12 bits on average, resulting an overall storage requirement of 55MB.

3.3 Improved Hardware Accelerator

3.3.1 Modified System Architecture Overview

We will now discuss how we implement separable beamforming as an extension to the previously discussed Sonic Millip3De beamforming accelerator. As discussed in Chapter II, Sonic Millip3De is a system architecture and accelerator for 3-D ultrasound that combines numerous hardware design techniques to minimize power while simultaneously generating high fidelity 3-D images. The full system comprises three distinct die layers (shown in Fig. 3.4) that are stacked vertically using modern 3-D

die-stacking techniques and are connected with through-silicon-vias (TSVs) [8]. This unique hardware layout allows for a dense, highly parallel design that can be easily integrated directly into the ultrasound scanhead, performing all front-end computation locally in a hand-held wand without the need for a large external system.

The first change to the hardware is the modification of the first die layer to support the reduced array size of the new sliding sub-aperture scheme. The new layer now uses a 120×88 array, over the previous 128×96, of capacitive-micromachined ultrasonic transducers (CMUTs) [62]. Using the new sub-aperture scheme for transmit and receive, these transducers are still multiplexed into 1,024 output channels, which are then fed into the second layer.

As before Sonic Millip3De's second layer is made up of ADCs and SRAM storage. For each of the 1,024 processing channels, there is a 12-bit ADC as well as a 6kB SRAM array to store the digital signal during the first stage of beamforming. Additionally, this layer now also has a secondary set of 1,536 6kB SRAM arrays which are used to store partially beamformed data for the second stage of separable-beamforming with data being fed back to this storage from the accelerator for the second pass. This second set of arrays is necessary to prevent the original echo data from being overwritten during beamforming as it is reused over a series of scanlines.

The final layer is the beamforming accelerator, which reads echo data from the SRAM arrays and generates beamformed output. However, to perform separable beamforming, data must pass through this layer twice, once for each partial beamforming operation. In the following section, we provide a more complete description of this layer and its modified operation below.

3.3.2 New Beamforming Accelerator

The beamforming accelerator is the central component of Sonic Millip3De, combining massive parallelism with a hardware-efficient implementation of the piecewise

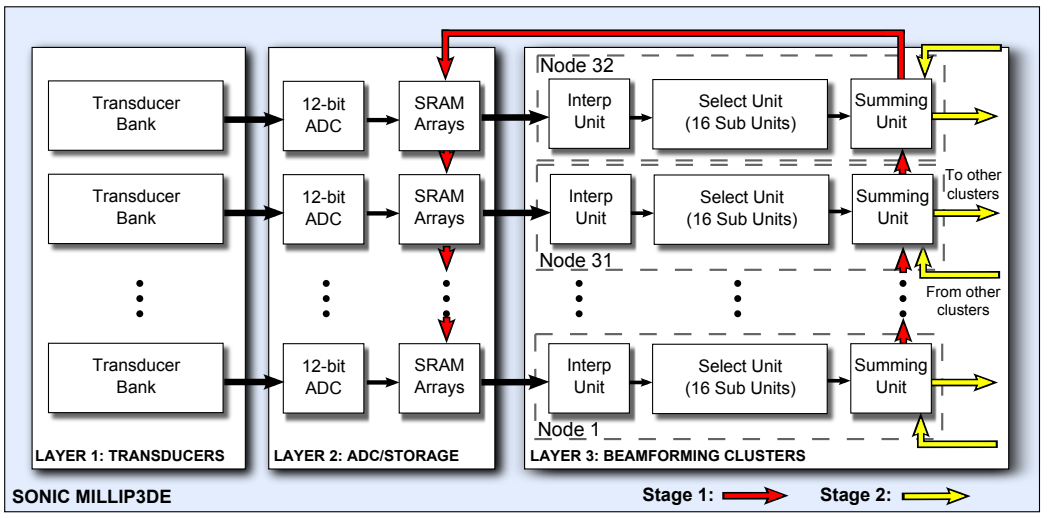


Figure 3.4: **Sonic Millip3De Hardware Overview.** The full hardware design is laid out over three distinct die layers connected vertically via TSVs. Layer 1 (24×18mm) comprises 120x88 transducers, with the analog transducer outputs multiplexed for each sub-aperture and routed over TSVs to Layer 2, comprising 1024 12-bit ADC units operating at 40MHz and SRAM arrays to store incoming samples. Data buffered in the SRAMs are transferred via face-to-face links to Layer 3 for processing in one of the 1024 3-unit pipelines of the beamsum accelerator. The interpolation unit upsamples the signal to 160MHz and performs apodization. The select unit maps signal data from the receive time domain to the image space domain. The summing unit combines the data across the 32 channels belonging to a particular cluster to construct the partial beamsum. The partially beam-formed data is transferred back to the SRAM layer to store until the second beamforming stage. In the second stage, the data are again sent through beamforming accelerator; however, the summing units are reconfigured to sum across all 1024 pipelines, arriving at the final 3-D image, which is then written to external memory.

quadratic approach to delay estimation. The accelerator comprises 1,024 parallel processing channels, which each read data from separate input channels and process 16 scanlines at a time. Each of these channels is further broken into a three-unit pipeline, which translates raw echo data stored in the SRAM layer into the beamformed data for the image (Fig. 3.4). During the first stage of separable beamforming, partial beamforming is performed within 32-channel clusters that perform a summation within the cluster and write partially beamformed data back to secondary SRAM storage in the second (memory) layer. The partially beamformed data is then fed through the accelerator a second time, where it is again delay-aligned and summed across all

1,024 channels to generate the final image. The image is then written to external memory via a ARM Cortex M-3 control processor.

As noted, each beamforming channel comprises three units. The first unit (interpolation unit) reads echo data from the SRAM storage and applies a pre-loaded channel-specific constant apodization to the signal. The apodization weights the channel's impact on the final image based on the corresponding transducer's position in the sub-aperture. After apodization, this unit then performs a 4× linear interpolation to up-sample the signal from 40MHz to 160MHz, a common optimization in existing commercial designs to reduce the ADC sampling frequency.

Next, the expanded data is streamed into the next unit for the beamformation process to begin. The interpolated signal is transferred from the interpolation unit to the select unit. The select unit iteratively calculates the delays between consecutive focal points along a scanline and identifies the interpolated sample that most closely corresponds to the focal point (i.e., it selects the sample from its channel nearest to each focal point). The select unit operates in parallel on 16 scanlines. 16 sub-units iterate over the interpolated data in a block-synchronized fashion each aligning the input signal to its assigned scanline. As described previously, the iterative delay calculation algorithm determines how many samples to advance an input channel to arrive at the sample nearest a focal point using the piecewise quadratic delay estimation formula. The hardware is easily able to estimate the delta between selected samples using three adders and the pre-computed quadratic constants, thereby iteratively solving the quadratic equation and producing each estimated delta as needed. Using these estimates, the sub-units know how far along the data stream to iterate before selecting their next output value. The delay-adjusted scanline data for the 16 neighboring scanlines is then fed forward to the summation network.

The final unit of each channel sums partially beamformed data across the channels though the use of adders connected via a reconfigurable mesh network. The new

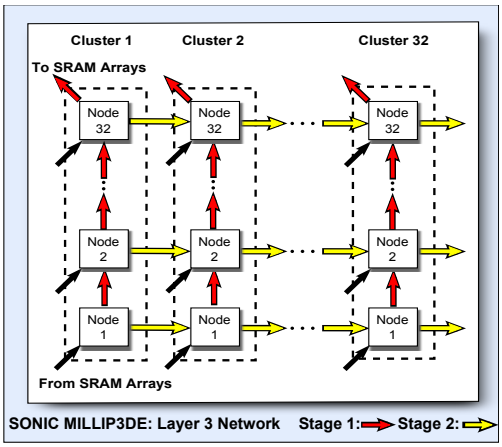
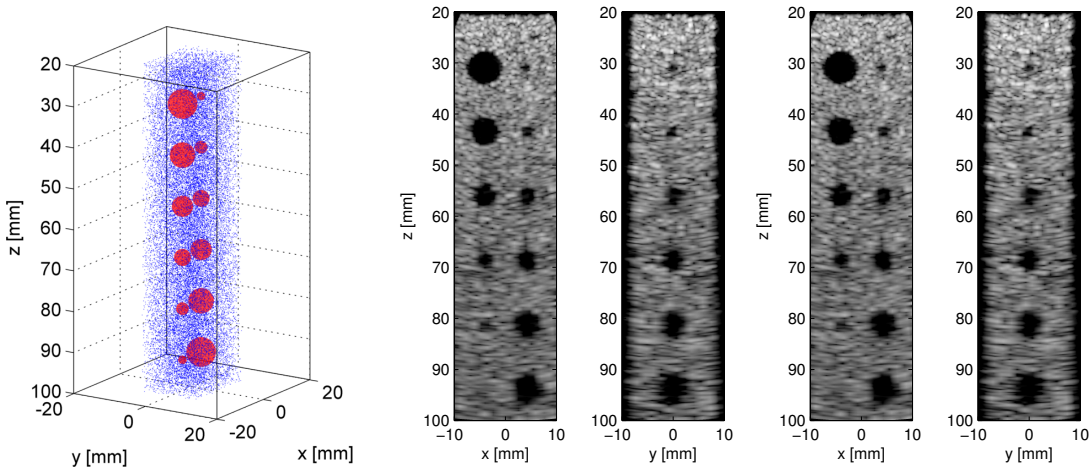


Figure 3.5: Network data flow in stages 1 & 2 of separable beamforming: The beamforming accelerator units in a dashed box form a cluster with the black arrows corresponding to data flow from SRAM arrays to nodes in a cluster. In the 1st beamforming stage the data is summed from bottom to top and is written back to the secondary SRAM arrays. During the 2nd stage, the network is reconfigured so that summation occurs from left to right, after which the fully beamformed data is stored in the DRAM.

mesh network design is reconfigured between beamforming operations to connect adders into a pipeline appropriate to the necessary summation operation: within clusters of 32 channels in the first beamforming stage, and across clusters in the second beamforming stage. The reconfigurability of the summation network is one of the key changes required over the baseline Sonic Millip3De design to enable separable beamforming.

The output of the summation network is written either to secondary SRAM arrays on the memory layer (for the first beamforming stage), or are passed to an ARM Cortex M-3 control processor to write final image data to external memory. (Fig. 3.5) illustrates the reconfigurable network and the data flow in the two beamforming stages.



(a) Simulated Phantom A **(b)** Non-separable Results **(c)** Separable Results

Figure 3.6: (a) Simulated phantom of tissue. Phatom has two rows of six anechoic cysts with diameters of 2 mm to 7 mm lying in the x plane. (b) 2-D slices from non-separable beamforming results. Average CNR is 1.99 and CR is 0.553. (c) Slices from separable beamforming results. Average CNR is 1.99 and CR is 0.549 showing that new method has comparable results.

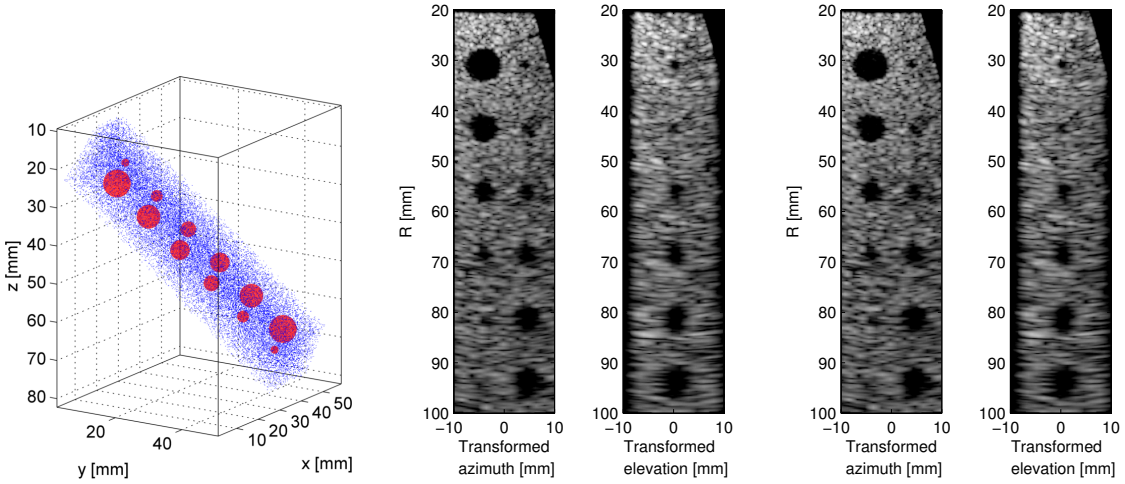
3.4 Separable Beamforming Simulation Results

3.4.1 Methodology

We evaluate image quality through simulated beamforming of cyst phantoms using Field II [41, 42] and MATLAB. The simulation parameters are listed in Table 3.1. The system employs a 2-D transducer array comprising 120×88 transducer elements with a central frequency of 4MHz and 50% fractional bandwidth. The scan view is 45° in both elevation and azimuth angles. The maximum depth of view is 10 cm.

3.4.2 Separable Beamforming

We consider two simulation cases, as illustrated in Fig. 3.6a and Fig. 3.7a. Both cases have twelve anechoic cysts located in a 20mm×15mm×80mm volume of random scatterers. The diameters of the cysts range from 2mm to 7mm. In Case A (Fig. 3.6a), the volume containing cysts and scatterers is vertical, corresponding to $\theta = \phi = 0°$. In Case B (Fig. 3.7a), the volume containing cysts and scatterers, with cysts located



(a) Simulated Phantom B **(b)** Non-separable Results **(c)** Separable Results

Figure 3.7: (a) Simulated phantom of tissue. Phatom has two rows of six anechoic cysts with diameters of 2 mm to 7 mm, but cysts are aligned with $\theta = \phi = 30°$. (b) 2-D slices from non-separable beamforming results aligned along 30° angles. Average CNR is 1.55 and CR is 0.552. (c) Slices from separable beamforming results, again aligned along 30°. Average CNR is 1.45 and CR is 0.545 showing again that new method has comparable results despite the large angle.

at $\theta = \phi = 30°$. Consequently, the field of scan view is increased from $45° \times 45°$ to $90° \times 90°$, and the number of scanlines is increased from 48×48 to 96×96.

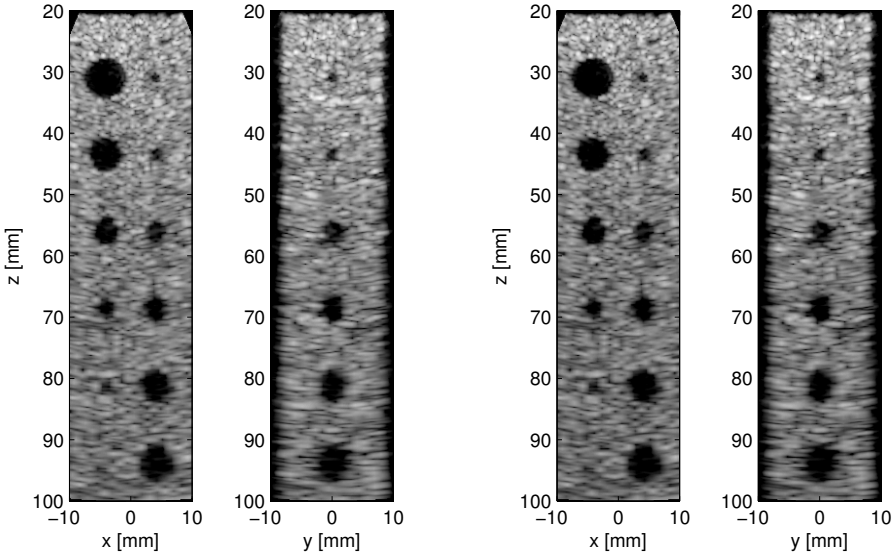
We quantify image quality via Contrast-to-Noise Ratio (CNR) and Contrast Ratio (CR). The CNR and CR are defined as follows

$$\text{CNR} = \frac{|\mu_{\text{cyst}} - \mu_{\text{bgnd}}|}{\sqrt{\sigma^2_{\text{cyst}} + \sigma^2_{\text{bgnd}}}} \tag{3.9}$$

$$\text{CR} = \frac{\mu_{\text{bgnd}} - \mu_{\text{cyst}}}{\mu_{\text{bgnd}} + \mu_{\text{cyst}}} \tag{3.10}$$

where μ_{cyst} and μ_{bgnd} correspond to mean brightness of cyst and background, and σ_{cyst} and σ_{bgnd} are the standard deviation of cyst and background.

The image quality of the 2-D slice images obtained in Case A by the baseline non-separable beamforming (shown in Fig. 3.6b) and the proposed separable beamforming method (shown in Fig. 3.6c) are nearly indistinguishable; both achieve an average



(a) 12-bit Separable (Phantom A) **(b)** 14-bit Separable (Phantom A)

Figure 3.8: (a) Slices from 12-bit separable beamforming for phantom A. Average CNR is 1.98 and average CR is 0.539. (b) Same as a, but with 14-bit data path. CNR is 1.99 and CR is 0.546.

CNR of 2.0 and an average CR of 0.55.

The 2-D slices of 3-D images obtained in Case B by non-separable beamforming and separable beamforming method are shown in Fig. 3.7b and Fig. 3.7c, respectively. We perform coordinate transformation and scan conversion in order to display the 2-D slices vertically. In these images, the vertical axis indicates depth R rather than the z axis coordinate. The images produced by non-separable method achieve an average CNR of 1.55 an average CR 0.55, while the images produced by separable method achieve an average CNR of 1.45 and an average CR of 0.55.

Finally, we confirm that the fixed-point performance of the proposed methods matches the quality of full double-precision floating point. We compare results of 12-bit and 14-bit separable beamforming in Fig. 3.8a and Fig. 3.8b. Both the 14-bit and 12-bit implementation achieve the same average CNR of 2.0, as in the double-precision-floating-point separable beamforming, although the 12-bit implementation has a slightly lower average CR compared to the 14-bit implementation (0.54 vs

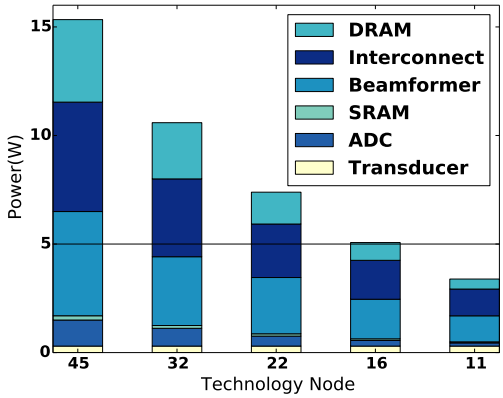


Figure 3.9: Power Breakdown Across Technology Nodes. Scaling projections based on trends reported in [23, 59]. We project meeting the 5W power budget at the 16nm node.

0.55). Compared to the 14-bit non-separable beamforming suggested in [75], 12-bit is sufficient for separable beamforming, because truncations can be done on partial beamforming data $F_l^{(1)}(n_y, m_R, m_\theta; t)$ to prevent overflow without affecting image quality. Hence we propose a 12-bit data path in our hardware implementation for separable beamforming.

3.4.3 Power Analysis

To analyze the power and performance of our separable beamforming system, we use the same power estimation techniques for each component of the design as we did in Chapter II. For the beamforming accelerator, we use RTL-level Verilog synthesis results of the accelerator hardware using an industrial 45nm standard cell library. SRAM values are generated using an industrial SRAM compiler, and our network power is obtained using SPICE models of our wires in 45nm. Published state-of-the-art power numbers are used for ADC[87], DRAM[55], and memory interconnect (ARM Cortex M-3)[6].

In addition to our power analysis at 45nm, we also project power requirements to 11nm technology using published trends. ADC scaling uses values from [59], technology scaling is taken from [23], and we assume network wire power does not scale

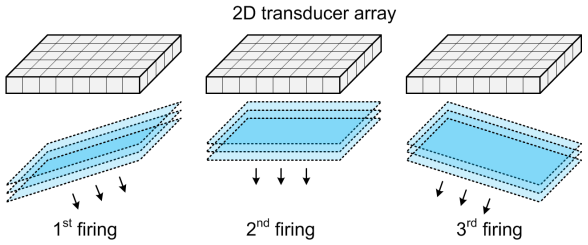


Figure 3.10: Firing scheme of 3D plane-wave system with compounding

other than the shortening of the wires due to transistor area scaling.

Fig. 3.9 shows the complete power breakdown of each component and a total system power at 45nm of just below 15W for a frame rate of 32Hz. Based on the scaling trends, Sonic Millip3De is now just within our 5W target at the 16nm node and falls well below the target power by the 11nm node. Thus, compared to our previous implementations in Chapter II, the separable beamforming method improves the frame rate from 1-2Hz to 32Hz without increasing the whole system power consumption.

3.5 Separable Beamforming with Planar Wave Imaging

Separable beamforming enables our Sonic Millip3De hardware to improve its frame rate by over 32×. However, some applications which do not rely on images directly, such as velocity estimation, require temporal resolution on the order of 1k volumes per second. Therefore the improvements we have previously discussed are not adequate for Sonic Millip3De to be used in these applications. In this section we discuss a modified transmission scheme that allows our hardware to achieve such a high volume rate.

3.5.1 Improved Sonic Millip3De Performance with Plane-wave Imaging

Plane-wave imaging is a technique that modifies the ultrasound transmission to generate images with fewer overall transmissions, allowing for higher frame rates to be achieved. This method, however, comes at the cost of some image degradation

and artifacting due to grating lobe effects [40].

During transmit of a plane-wave system, all transducers in the selected aperture fire in unison (or with a linear delay) and emulate a plane wave that is parallel (or angled) with respect to the transducer plane (shown in Figure 3.10). All the elements in the selected aperture receive the echo signals, and beamforming is performed in a similar manner as before. My collaborator Ming Yang extended the previously described separable beamforming to 3D planar transmissions and details regarding the methodology and specific beamforming calculations can be found in [94, 97].

The Sonic Millip3De hardware, using plane-wave transmissions and the 3D separable beamforming is able produce a much higher volume acquisition rate (greater than 1,000 volumes per second) with only a minor change to the system hardware. This increase is due to the simplicity of the separable planar imaging technique as well as a reduction from 96 to 9 firings for a single volume without modifying the underlying algorithm.

In our previous system design, sub-volume data from each firing was temporarily stored in off-chip DRAM before being combined to produce the final volume; however, due to the extremely high rate that these sub-volumes are produced for the planar technique (over 9,000 sub-volumes per second), bandwidth to off-chip DRAM is insufficient for temporary storage. To remove this bottleneck, we have modified the off-chip design to include an additional 4th die layer of embedded DRAM (eDRAM) to handle the temporary storage of the 21MB sub-volumes locally. Furthermore, we can avoid the refresh power conventionally required for DRAM since sub-volumes are overwritten so rapidly that there is no need to refresh them.

3.5.2 Planar Wave Imaging Simulation Results

We evaluate image quality in MATLAB using Field II [41, 42]. The system parameters are listed in Table 3.2. The baseline system employs non-separable beamforming

method, and the plane wave system uses compounding from nine plane waves, each at a different angle with respect to the transducer plane.

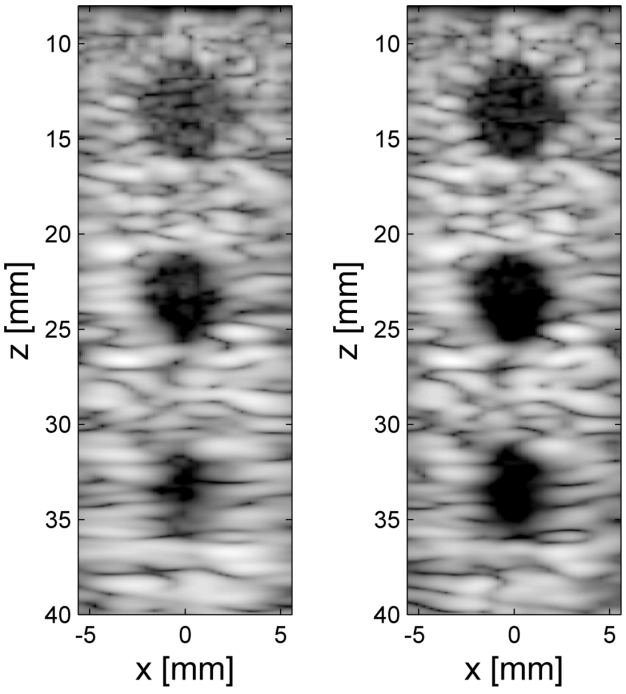
Table 3.2: System parameters of the 3D plane wave system

Property	Value
Pitch, μm	385
Receive aperture size, transducers	32×32
f-number	2.0
Number of scanlines	32×32
Max depth, cm	5
Center frequency, MHz	4
6 dB transducer bandwidth, MHz	2
A/D sampling rate, MHz	40

Three 6mm anechoic cysts located in phantom scatterers at depths of 12 mm, 23 mm and 33 mm are simulated. The CNR values provided by the non-separable beamforming without compounding are 1.8, 1.9 and 1.2, respectively; the separable beamforming with compounding improves the CNR values to 2.5, 2.4 and 1.7, respectively. The xz slices of the 3D volume are shown in Fig. 3.11. As the images show, the image quality of using plane-waves in combination of separable beamforming with coherent compounding method allows our hardware to achieve much higher image quality compared to a roughly $4\times$ slower imaging rate using no compounding and non-separable beamforming.

3.6 Conclusions

In this chapter, we have presented multiple techniques used to improve and expand the capabilities of our Sonic Millip3De hardware. We introduced the sliding sub-aperture firing scheme as well as a separable beamforming method that reduces the computational complexity of 3-D ultrasound imaging systems. The method is based on decomposing the delay term in a way that minimizes RMS phase error. We also proposed extensions to the Sonic Millip3De beamforming hardware accelerator to efficiently implement the separable beamforming method. Synthesis targeting an



(a) Nonsep, No Compound

(b) Sep, Compound

Figure 3.11: **Comparison of Plane Wave Methods:** (a) Non-separable plane wave beamforming without compounding. Cyst CNR from top to bottom: 1.76, 1.91, 1.20. (b) Separable plane wave beamforming using compounding from 9-angles giving considerable improvement in CNR: 2.46, 2.40, 1.70.

industrial 45nm standard cell library indicated that the design can produce 3-D images at 32Hz frame rate within a 15W power budget, compared to 1-2Hz frame rate in Chapter II without increasing the power consumption. As before we validated image quality via Field II cyst phantom simulations, which show that the proposed separable beamforming synthetic aperture ultrasound system can produce high quality images that are comparable to those produced by the non-separable baseline method.

We have also described Sonic Millip3De's performance when using separable beamforming in conjunction with coherent image compounding for 3D ultrasound plane-wave imaging systems. While separable beamforming reduces the beamforming computational complexity for each volume by about 11×, the results show that with a 9-fire-angle compounding scheme, the accelerator can achieve CNR of 2.2 at volume

acquisitions rates over 1000 volumes per second. This performance improvement enables applications that require high volume rates, such as velocity and volumetric flow estimation.

CHAPTER IV

MRI and X-Ray CT Overview

In this chapter we will give a brief overview of the other two primary forms of medical imaging: MRI and X-ray CT. We will begin by giving background on the basic algorithms used for each, and then we will cover the state-of-the-art techniques we are focusing on for workloads. After covering each of the modalities, we will discuss the results of our previous work [78] that shows that existing hardware solutions are inefficient and limit potential speedup gains.

4.1 Magnetic Resonance Imaging

4.1.1 Nuclear Magnetic Resonance

The basis of all magnetic resonance imaging (MRI) is the physical phenomenon of nuclear magnetic resonance (NMR). Fundamentally NMR is the process of electromagnetic excitation and relaxation of certain atomic nuclei in a magnetic field. The phenomenon is due to the intrinsic magnetic moment of nuclei; therefore, it only occurs in isotopes whose nuclei have such properties, specifically those with an odd atomic number (number of protons) or with an odd mass number (number of protons and neutrons). Under these conditions, the nucleus has an angular momentum or *spin* which in turn creates a nuclear magnetic moment, making such isotopes behave like magnetic dipoles and respond to magnetic fields.

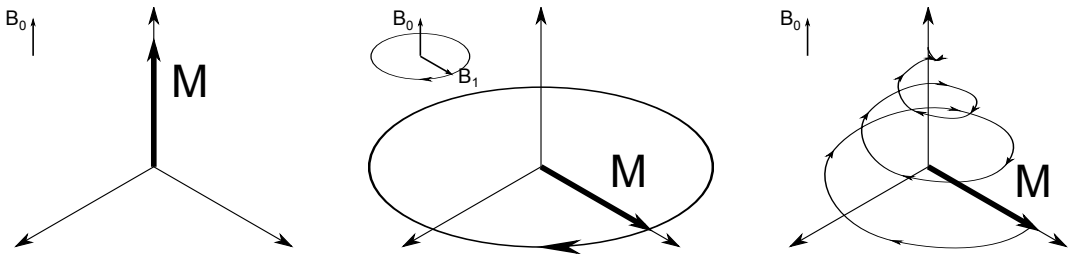
By placing the nuclei in an external magnetic field $\boldsymbol{B_0}$, two effects occur. Assume $\boldsymbol{B_0}$ is aligned along z such that $\boldsymbol{B_0} = B_0\hat{\boldsymbol{z}}$ where $\hat{\boldsymbol{z}}$ is a unit vector in the positive z direction. First, a slight majority of magnetic dipoles will attempt align into a steady-state with $\boldsymbol{B_0}$. This effect will create a magnetization vector $\boldsymbol{M}$ that is the sum of all of the dipoles along z (shown in Fig. 4.1a). Second, the force of $\boldsymbol{B_0}$ in conjunction with the angular momentum of the nuclei will cause an additional torque on the nuclei. This torque with the alignment along $\boldsymbol{B_0}$ causes the nuclei to wobble or *precess* around $\boldsymbol{B_0}$ instead of perfectly align. Additionally the frequency of the precession, known as the *Larmor frequency*, is completely determined by the magnitude of $\boldsymbol{B_0}$ and is given by the equation

$$\omega_0 = \gamma B_0 \tag{4.1}$$

where γ is the *gyromagnetic ratio* constant and is specific to each isotope.

Once the nuclei are in a steady-state alignment with z and precessing, they can now be excited through the introduction of a second magnetic field $\boldsymbol{B_1}$ that is perpendicular to $\boldsymbol{B_0}$. However, the goal of this excitation is to move $\boldsymbol{M}(t)$ into oscillation around the transverse (xy) plane while still maintaining the precession around z; therefore, a static $\boldsymbol{B_1}$ cannot be used as it would simply move the precession (and therefore $\boldsymbol{M}$) around a new axis. Instead, $\boldsymbol{B_1}$ is applied at the Larmor frequency as a circularly polarized RF excitation in the transverse plane (i.e., the tip of the magnetization vector traces a circle in the xy plane over time).

The length and amplitude of this excitation can produce different effects, but it is primarily used in two ways. First, is an *excitation pulse* which causes the previously described effect of moving $\boldsymbol{M}(t)$ into the transverse plane. This effect occurs not because the individual dipoles themselves have rotated into this plane, but instead because approximately half of them have been excited into a higher energy state



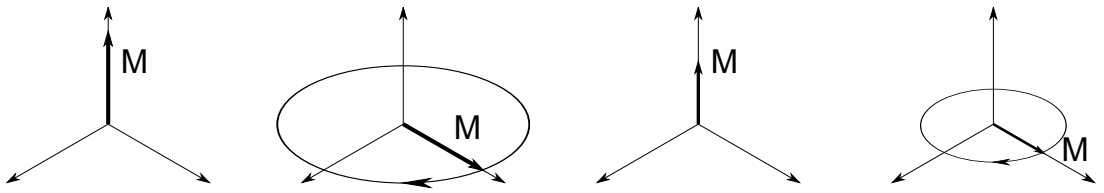
(a) Equilibrium **(b)** Excitation **(c)** Relaxation

Figure 4.1: **Excitation and Relaxation:** (a) Magnetization vector $\boldsymbol{M}$ in equilibrium, aligned with $\boldsymbol{B_0}$ along z. (b) $\boldsymbol{M}$ after excitation pulse $\boldsymbol{B_1}$ Longitudinal excitation causes the z component to be cancel out and alignment of precessions has created a transverse component. (c) Relaxation back to equilibrium after $\boldsymbol{B_1}$ is removed. z component returns because of longitudinal relaxation (Eq. 4.2) while transverse component dies out due to dephasing of precessions (Eq. 4.3).

where they have flipped along z. This causes the z components overall to cancel each other out and $\boldsymbol{M}$ to go to 0 in z. Additionally all of nuclei's precessions have been excited into phase together giving $\boldsymbol{M}$ a significant xy component that rotates with the precession (shown in Fig. 4.1b). The other excitation effect occurs when the excitation pulse is doubled in length, all of the dipoles will flip, creating an *inversion pulse* and giving $\boldsymbol{M}$ a negative z component. While this effect may not appear immediately useful, it can be used in conjunction with excitation pulses to create another effect known as *spin echoes* that we will discuss later.

After the nuclei have been excited, $\boldsymbol{B_1}$ is turned off and $\boldsymbol{M}$ can be measured. This is a relatively simple process of measuring the current induced by the magnetic fields in neighboring coils. The process of returning to steady-state is not immediate though, and $\boldsymbol{M}$ will instead spiral back into alignment with z over a short period of time. This return has two decay components known as T_1 and T_2. T_1 comes from the time for all of the dipoles to flip back to their equilibrium state along positive z, known as *longitudinal relaxation* and given by the equation

$$M_z(t) = M_0(1 - e^{-t/T_1}) + M_z(0^+)e^{-t/T_1} \tag{4.2}$$



(a) Equilibrium **(b)** First Excitation **(c)** Partial Relaxation **(d)** Second Excitation

Figure 4.2: **Measuring T_1 Relaxation Effects:** (a) $\boldsymbol{M}$ in equilibrium. (b) After first RF excitation. (c) $\boldsymbol{M}$ after full transverse relaxation, but only partial longitudinal relaxation. Note that amplitude of this vector will be directly dependent on T_1 value. (d) Second excitation causing $\boldsymbol{M}$ to tip into transverse plane again; however, this magnitude will be again dependent on T_1 allow for T_1 effects to be measured.

where $M_z(0^+)$ is the longitudinal magnetization immediately after the $\boldsymbol{B_1}$ is turned off and M_0 is the steady-state magnetization in $\boldsymbol{B_0}$. The second relaxation component T_2 is based on the *transverse relaxation* which is the dephasing of the precessions in the transverse plane. The transverse relaxation is given by

$$M_{xy}(t) = M_{xy}(0^+)e^{-t/T_2} \tag{4.3}$$

where $M_{xy}(0^+)$ is similarly the transverse magnetization an order of magnitude shorter than T_1.

Also in practice both T_1 and T_2 are difficult to measure directly. Because T_1 is a longitudinal effect, it cannot induce measurable current in the coils that run parallel to the patient. Instead a multi-pulse excitation method is used to first align M along the transverse plane, allow for a full T_2 decay but partial T_1 decay, and then re-excite to align M along the transverse plane again (shown in Fig. 4.2). Before the second excitation occurs the xy component of M will be gone, but only part of the z component will have been restored. Therefore during the second excitation, only nuclei which returned to their lower energy state will be excited. Because of this M will have a reduced amplitude in xy that corresponds to the amount of nuclei that were able to return to their resting energy, creating an measurable T_1 effect.

The difficulty in measuring T_2, on the other hand, is due to another value known as T_2^* which is the observed transverse relaxation constant. T_2^* decays significantly faster than T_2 and is given by

$$1/T_2^* = 1/T_2 + 1/T_2' \tag{4.4}$$

where T_2' is the decay caused by experimental effects such as inhomogeneities in the magnetic field. Because of these other effects, T_2 can be difficult to isolate. This is where the previous mentioned spin echoes come into effect. When a T_2 weighted image is desired, the signal is pulsed as usual using an excitation pulse. After a brief period t, an inversion pulse is applied causing a complete flip of M. After another period of t the phases will realign without the effects of T_2' (since the flip will cause only the natural dephasing to reverse) allowing for T_2 effects to be measured.

Both T_1 and T_2 depend heavily on the atomic composition of the material. In MRI this translates to different signal strengths of different tissues after a given amount of relaxation time. Because of this, MRI images can be weighted differently based on amount of relaxation time and excitation method before measurements are taken: proton density (no relaxation time with single excitation), T_2 (small relaxation time with spin echoes), T_1 (small relaxation with pulsed excitation). Additionally contrast agents can be applied to further modify the relaxation times to produce additional methods of contrast between tissues.

4.1.2 MRI: Creating an Image from NMR

As discussed, NMR is only possible for isotopes whose nuclei have spin. Luckily humans have one such isotope in very large quantities: hydrogen (^{1}H). Because of the large amount of water and hydrocarbons throughout the human body, E-M effects produced through NMR of hydrogen create high signal strength and allow for high

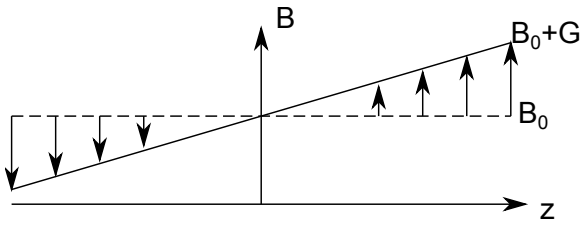


Figure 4.3: Magnitude of magnetic field along z when linear slice gradient G is applied. For a given excitation frequency, only areas with the appropriate magnitude of B will be excited where B is given by Equation 4.1. Frequency and phase gradients are applied in a similar manner during other stages of the MRI process to obtain further spatial encoding.

resolution imaging of soft tissue types. For this reason hydrogen is the focus of nearly all diagnostic MRI.

With the isotope of interest and corresponding gyromagnetic ratio γ fixed, one can calculate the correct Larmor frequency based on the initial magnetic field B_0 (in clinical use, B_0 is typically 1.5 or 3 Tesla) of the main magnet. After the patient is in the constant magnetic field and hydrogen nuclei are in alignment, the magnetic field can be manipulated via gradient coils for x, y, and z. These coils allow magnetic gradients across the main magnetic field B_0 during different parts of the excitation/relaxation process to create a spatially dependent magnetic field.

The first type of gradients are known as *slice gradients*. Slice gradients are typically applied along z (z here is the standard MRI orientation along the length of the patient, increasing towards the feet) during steady state to create a new initial magnetic field $B(z) = B_0 + G_z z$ (Figure 4.3). Because the Larmor frequency is set by B it will now vary along z; therefore, an excitation of a specific frequency will only excite a small segment (or slice) along the patient. By varying gradient slope, G_z, the width of the slice can be increased or decreased to best fit the region of interest.

The second type of gradients are known as *frequency gradients*. Frequency gradients are applied to the magnetic field immediately before the NMR signal is read in to change the precession frequency of the already excited nuclei. Now when the signal is captured via the induced current, the signal can be spatially decoded via Fourier

analysis based on the frequency gradient. By using different gradients over different excitation sequences, further spatial information can be achieved. In practice, frequency gradients are typically used along x, y, or both depending on the method of image generation.

The final type of gradients are known as *phase encoding gradients*. Similar to frequency gradients, the purpose of phase gradients is to encode information spatially that can be extracted from the captured signal. However, phase gradients achieve this by encoding phase information within the data of the same frequency. This process can be achieved with very short gradient bursts after excitation but before the signal is captured. Due to the nature of the phase gradient, it is typically done perpendicular to the frequency gradient, such as along y for an x frequency gradient or along z to provide spatial information within the slice.

Using different gradient sequences in conjunction with Fourier analysis, full 3D spatial resolution can be acquired producing the final MRI image. However, generating images that are combinations of many sequences can introduce motion artifacts. Additionally, many modern MRI applications want to track changes over time (dynamic MRI), requiring even fewer firings. To handle these cases, modern MRI research focuses on undersampling spatially and using additional techniques to reconstruct images from incomplete data.

4.1.3 Dynamic MRI with Compressed Sensing using L+S and Golden-Angle Radial Sampling

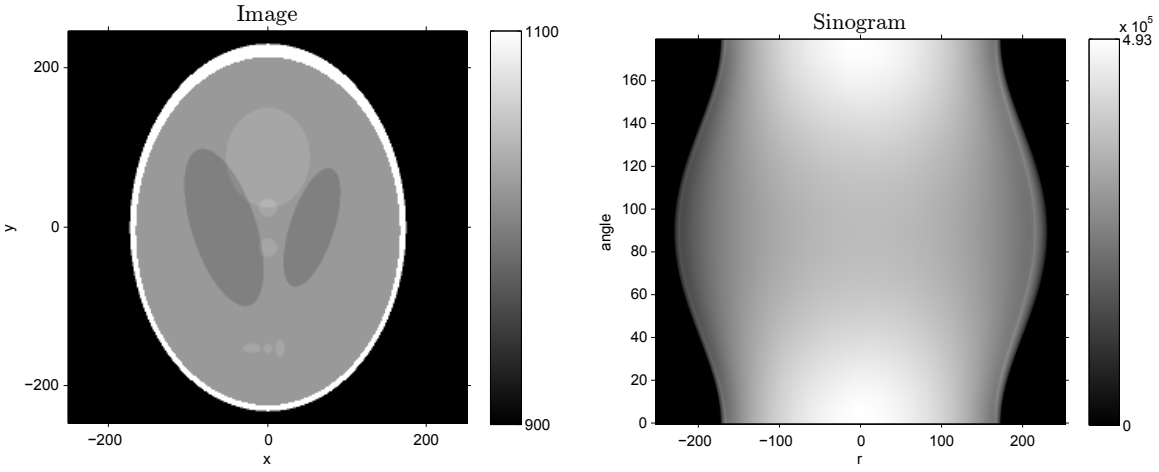
Basic MRI image reconstruction is not a difficult problem, and modern computer hardware can easily compute the Fourier transforms necessary to generate the image. However, many modern dynamic MRI applications require large imaging volumes to be generated at such high frame rates that complete spatial information simply cannot be physically acquired fast enough. This has led to an entire area of MRI research

which uses compressed sensing (CS) techniques with heavy computational burdens to create acceptable images, despite spatially under-sampling. Much of this work is still under development and can be further improved with more computational power which is why we believe it will be an important part of the medical imaging benchmarks in MIRAQLE. In particular our MRI workload will focus on a combination of golden-angle radial sampling [90] in conjunction with the low-rank plus sparse matrix decomposition technique [25, 63] as these methods combine many state-of-the-art techniques, and the code used for these reconstruction algorithms is freely available.

The imaging process itself is done using golden-angle radial sampling [90], a method that consists of rotating xy plane gradients by an angle of 111.246° between each data capture. The golden-angle technique provides flexible temporal resolution for dynamic MRI by allowing for uniform angular sampling for various window sizes that are Fibonacci numbers (2, 3, 5, 8, etc.). Additionally, because our MIRAQLE workload makes use of radial sampling (via golden-angle), it also incorporates advanced non-uniform FFT (NUFFT) [80] methods for image reconstruction.

In addition to radial sampling, we also implement the L+S algorithm which is a further refinement of existing CS methods. L+S begins by creating an initial matrix M from transformed receiver signals by representing each temporal frame as a column. Then attempting to fill in gaps and de-noise using a minimization problem. However, at each step of the iteration, before minimizing the residual, L+S goes one step beyond other CS methods by decoupling M into a low-rank matrix (L) which has few non-zero singular values and a sparse matrix (S) which has few non-zero values overall and minimizing these values independently. By doing this, L+S allows slow background changes (L) to be extracted and fitted separately from fast movement and noise (S). A full description of this algorithm can be found in Table 1 and Figure 3 of [63].

Overall this technique has shown great promise in allowing for a significant scan



(a) Original Phantom **(b)** Sinogram

Figure 4.4: **Sinogram:**(a) Synthetic 2D phantom that is being imaged. (b) Sinogram created from 492 different views (angles).

acceleration over previous CS methods allowing for high resolution dynamic images with even less temporal sampling. However, the decomposition of the method creates even higher computational demand, limiting its current use in the clinical setting, and making it an excellent candidate for MIRAQLE.

4.2 X-Ray Computed Tomography

4.2.1 Filtered Back Projection

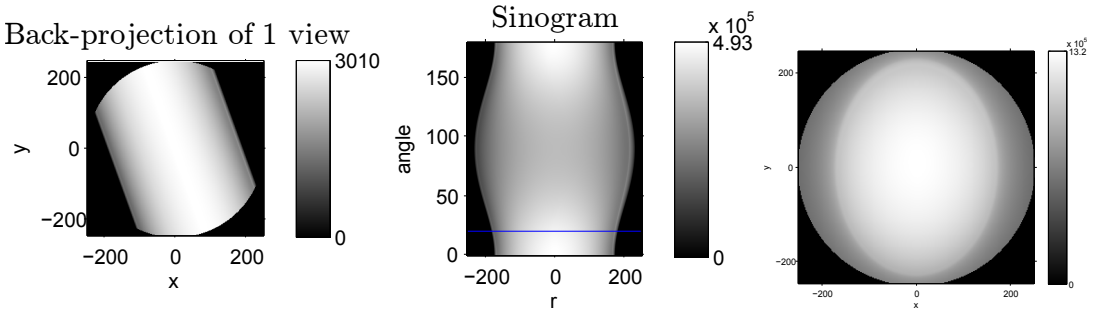
X-ray computed tomography (X-ray CT) is the process of taking several X-ray images (either 1D or 2D) at different angles and combining them into one final image (2D or 3D). The process of creating an image from the received X-ray signals can be done using several different techniques. However, we will begin by introducing the simplest and, until recently, most common method of creating this image, a process known as *filtered back projection* or FBP.

In a basic 2D example of filtered back projection, parallel X-rays are sent through a patient from one side of the body to the other and then finally to a row of detectors on the other side. This produces a 1D X-ray image of data from the detectors where

each data value corresponds to a specific ray from the emitter, through the patient, to the given detector. Each value in the "image" represents the amount of X-ray that was able to pass through the patient for that ray. In other words, the value represents the line integral of all attenuation through the various tissues that the ray intersected. This process is then repeated over several different angles (0 to just under π is necessary to get all possible angles) to obtain an array of different ray paths. Additionally all of the 1D slice images can be concatenated to generate a *sinogram*, a 2D representation of the projections where one axis corresponds to the receivers and the other axis corresponds to the angle (Figure 4.4). While sinograms are not immediately useful to a radiologist, they serve as a visual representation of the raw X-ray data received through forward projection. The idea of the sinogram also makes it easy to conceptualize the reversal of the process via a back projection into an image.

To reverse the process of converting the sinogram back, each 1D slice is projected back into the 2D image space along each ray that the line integrals represent producing a *laminogram* shown in Fig. 4.5a. The laminogram is effectively a back projection of a single transmission. All of the laminograms can then be summed together to produce an image, resulting in a complete (non-filtered) back projection. However, while this is close to the desired image, the back projection results in a significant amount of blur (as seen in Fig. 4.5b), especially where there is a lot of signal, such as in the patient. This blur is a mathematical byproduct of the back projection process caused by the fact that mathematically reversing the projection is not equivalent to inverting it.

To remove the blurring effect, a high-boost type filter is applied to the sinogram before the back projection occurs. This filtering allows for proper recovery of the desired image. The simplest filter method is to use a ramp filter which has a value of 0 at 0 frequency and then scales linearly upward as frequency increases (or decreases)



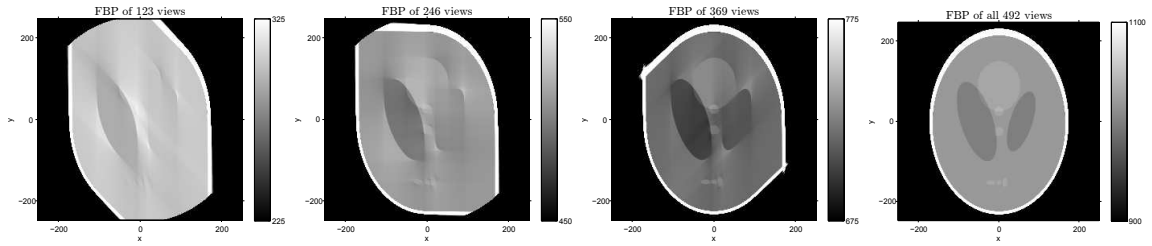
(a) Laminogram **(b)** Full Back Projection

Figure 4.5: **(Unfiltered) Back Projecting:** (a) Unfiltered back projection of single view (laminogram) and the slice of sinogram being back projected. (b) Full back projection from all views, dominated by blurring.

from 0. This causes high frequency (i.e. sharp changes in neighboring line integrals) to be accentuated and sharpened, removing the back filter blurring effect. With the data properly filtered, the final step is to take the filtered sinogram and perform the previously described back projection to obtain the final FBP image result. Figure 4.6 shows the result of each quarter step of the final summation.

However, this description of FBP assumes that all of the transmission rays are parallel and are received by a linear array of receivers for a given fixed angle. While this is an accurate model of the earliest CT systems, modern CT systems have moved away from this design. Today fan-beam or the 3D equivalent, cone-beam, use a single transmission source location and a curved 1D or 2D array for receivers. Moving to this model does change the intersection pattern; however, the FBP method can be modified to account for this.

Overall FBP is one of the most straight-forward and basic reconstruction techniques for CT, but it is a fairly simplistic model with many flaws. The FBP method makes simple assumptions about X-ray attenuation and does not account for noise or movement which has lead to the development several other approaches. However, these newer techniques, discussed below, come at significant computational costs.



(a) 1/4 **(b)** 1/2 **(c)** 3/4 **(d)** Full

Figure 4.6: **Filtered Back Projection:** Summation of filtered data that is back projected from (a) 1/4 views, (b) 1/2 views, (c) 3/4 views, (d) all views.

4.2.2 Algebraic Reconstruction Technique

A completely different approach to CT reconstruction is the algebraic reconstruction technique (ART) [33] which converts the image reconstruction process into a discrete linear algebra problem. By using a linear algebra approach, the solution can be achieved through an iterative process that is robust to imperfect data than simply performing FBP; however, this technique does come at the cost of much more computation and introduces new convergence concerns that must also be carefully addressed.

Recall that CT data consists of multiple line integrals representing attenuation for a specific ray through the patient. To discretize this model, the image space is divided into pixels, or in the case of 3D, voxels where each voxel is assumed to have a uniform (but unknown) attenuation value. The collection of the p voxels is then represented as a vector $\boldsymbol{x}$ where $\boldsymbol{x} = [x_1, x_2, ..., x_p]^T$. Using this representation each line integral can now be described as a discrete summation across all of the intersected voxels in $\boldsymbol{x}$. Mathematically this can be represented as a summation across all of the voxels in $\boldsymbol{x}$ multiplied by weighting constants a_{ij} where i represents the specific ray and j goes from 1 to p. In a simple model a_{ij} could be a binary 1 if intersection occurs and 0 if otherwise, but the method is improved in practice by scaling a_{ij} based on the amount of intersection with a pixel, creating a more exact representation of the intersection. Putting all of the a_{ij} values together gives the system matrix

where $\boldsymbol{Ax} = \boldsymbol{y}$ produces the projection in the form of $\boldsymbol{y}$ which is the collection of line integrals (i.e. the sinogram). Using this representation, ART uses an iterative least-squares method to solve $\boldsymbol{Ax} = \boldsymbol{y}$.

Additionally, it should be noted that with this method of representation, the (unfiltered) back projection can easily be obtained by applying the transpose of $\boldsymbol{A}$ ($\boldsymbol{A}^T\boldsymbol{y}$). To remove the blur caused by $\boldsymbol{A}^T\boldsymbol{y}$, the inverse of the blur can be applied to give $[\boldsymbol{A}^T\boldsymbol{A}]^{-1}\boldsymbol{A}^T\boldsymbol{y}$ which is a matrix representation of the filtered back projection process.

4.2.3 Model-based Iterative Reconstruction

Unlike ultrasound and MRI, X-ray CT exposes the patient to potentially harmful radiation to obtain the image. The previously discussed methods, while able to produce high quality images, require a substantial amount of high-dose radiation firings to make up for their lack of robustness. However, state-of-the-art systems have attempted to lower the radiation through fewer and less-powerful firings which ultimately results in noisy and under-sampled data. Model-based iterative reconstruction (MBIR) [83] is one such approach that starts with a similar linear model to ART and incorporates more information about the object to compensate for missing data. This approach produces images comparable to those generated with higher dosage imaging [32]. However, this method is very computationally expensive, and it can be limited on even the most modern hardware. For our CT workload, we have chosen to incorporate a type of MBIR that is comparable to what is used in commercial systems today.

MBIR is very similar in concept to ART; however, the minimization problem uses statistical weighting to account for receive data reliability and an edge-preserving regularizer to add information on how an image should look, namely sharpening

edges and blurring non-edges. The MBIR problem can be stated as

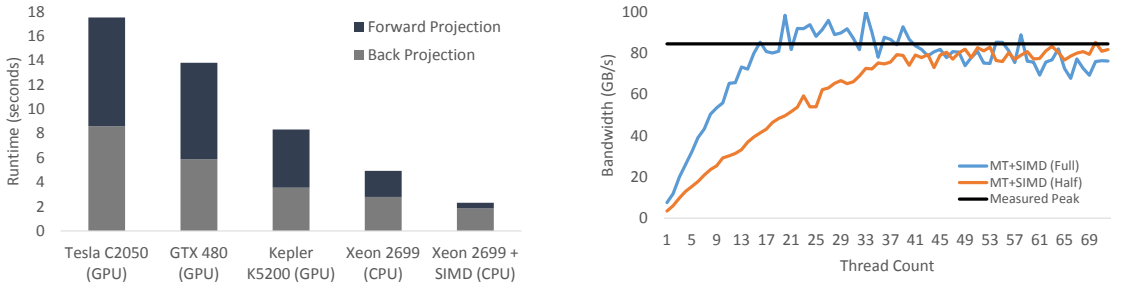
$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}\geq\boldsymbol{0}} \Psi(\boldsymbol{x}), \qquad \Psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2_{\boldsymbol{W}} + R(\boldsymbol{x}) \tag{4.5}$$

where $\boldsymbol{A}$ is the system matrix, $\boldsymbol{y}$ is the received data, $\boldsymbol{W}$ is the statistical weighting, $\|\|^2$ is the L^2 norm, and R is the regularizer. The particular method we have chosen to focus on uses a primal gradient decent based technique to find a solution. The algorithm updates $\boldsymbol{x}$ each iteration using a diagonal majorizer multiplied with a gradient of Ψ. The gradient of Ψ is calculated as the gradient of the regularizer plus a weighted back projection of the difference between y and the forward projection of the current x and is represented as

$$\nabla\Psi = \nabla R(\boldsymbol{x}^{(n)}) + \boldsymbol{A}^T\boldsymbol{W}(\boldsymbol{A}\boldsymbol{x}^{(n)} - \boldsymbol{y}) \tag{4.6}$$

where $\boldsymbol{x}^{(n)}$ is the reconstruction of $\boldsymbol{x}$ for the current iteration, n. However, speed up the computation the gradient calculation above is estimated by working on subsets of $\boldsymbol{A}$ and $\boldsymbol{W}$ at a time. More information regarding this technique can be found in [22, 50].

Additionally the technique we will be using for this work makes use of the distance driven (DD) method [20] to estimate the cone beam intersections during forward and back projection. In DD, the system matrix values are calculated on-the-fly through a process of estimating shadows cast by the ray for each voxel to compute the forward and back projections (the most computationally expensive components of MBIR process). During this process, the intersections of the rays with the edges of each voxel in a given row is projected onto a line. Then a similar process back projects the detector boundaries (as though they originated from the beam source) onto the same line. Using the overlap of the voxel shadow and detector footprint, the system matrix weighting can be computed for each voxel and detector pair, including fractional val-



(a) Gradient Runtime Comparison **(b)** Average Memory Bandwidth Consumption

Figure 4.7: Results reproduced from [78]. (a) Comparison of forward and back-projection runtimes on various GPU generations and on the dual-socket Xeon 2699. Xeon MT and MT+SIMD reconstructions use all 72 logical cores. SIMD implementation uses 8-wide floating-point AVX2 instructions. (b) Calculated bandwidth consumption during system matrix calculation during back-projection with multi-threaded SIMD for 1-72 threads, each averaged over 25 runs, compared with measured peak bandwidth of the system. "Full" denotes 32-bit single-precision data, and "Half" is emulated 16-bit precision by reading/writing half of the data. Peak bandwidth measured using STREAM triad benchmark[57] with 72 threads.

ues for partial overlaps. In particular, the DD technique has been shown to be highly parallelizable both for CPU and GPU implementations, leading to drastic improvement in MBIR runtimes [92]. However, modern hardware is still inefficient, leaving much of the potential of these algorithms unrealized.

4.2.4 Limitations of Existing Hardware

In our previous work [78] we explored parallelization opportunities of a very similar MBIR method using SIMD. The results of that study (shown in Fig. 4.7a) showed that 8-wide AVX2 single instruction multiple data (SIMD) programming plus multi-threading on current hardware improved performance over previous GPU techniques [58, 92]. However, we also found that the overall SIMD performance was saturated after only 25 threads, due mostly to bandwidth limitations (Figure 4.7b). These types of bandwidth restrictions are just one bottleneck of this type of computation, and with even more parallelism possible in up-coming hardware (such as 16-wide AVX-512 extensions), the performance lost due to inefficiencies will only continue.

CHAPTER V

MIRAQLE: Medical Image Reconstruction Algorithms and QuaLity Evaluation Benchmark Suite

In this chapter we discuss MIRAQLE[1], our newly developed medical imaging benchmark suite. MIRAQLE consists of three primary imaging tasks, 3D ultrasound beamforming, iterative reconstruction X-ray CT, and dynamic MRI reconstruction, with the goal of providing the computer architect community the tools needed to study these problems and develop improved architectural support for medical imaging. MIRAQLE not only contains workloads for study, but it also provides the necessary tools to perform basic image quality analysis to allow for algorithmic modifications (similar to those we used in Chapter II and III for Sonic Millip3De). In the first section, we revisit each of the imaging tasks, providing short backgrounds and discussing the specific imaging algorithms we have chosen. For each task we also discuss the input data we have generated as well as the quality metric we provide to evaluate the output. The next section demonstrates our image quality evaluation technique, showing potential image quality trade-offs that are possible. Finally we discuss architectural design opportunities for each modality that show promise for future hardware designs to explore.

[1] Available at miraqle.eecs.umich.edu

5.1 MIRAQLE

We first describe the imaging tasks that comprise MIRAQLE. For each task, we provide brief background on the imaging modality, discuss the image reconstruction algorithm, describe the imaging task, present the metric we use to quantitatively evaluate image quality, and discuss architectural challenges and opportunities.

5.1.1 Ultrasound Imaging

Background. MIRAQLE includes a synthetic aperture brightness-mode (B-mode) ultrasound imaging task. Synthetic aperture ultrasound imaging is a three step process of transmission, reception, and beamforming. During transmission, high-frequency (1-15 MHz) sound waves are generated by exciting one or more transducers using electrical current. Ultrasonic pulses travel as spherical compression waves into the patient, with the overall transmission being shaped, steered, and focused by co-ordinating the excitation of the transmit transducer.

As the signal encounters tissue boundaries, a partial reflection of the sound wave occurs, returning an echo of the original signal to the transducers. Highly reflective (hyper-echoic) regions produce high amplitude echoes, resulting in brighter areas in the image, while regions of low reflectivity (anechoic) produce weaker signal and darker areas. As the echoes return, the signals produce vibrations in the transducers, creating measurable electrical current.

Once enough time has passed for a complete round trip through the image space, the image reconstruction process (beamforming) can be performed. Beamforming is typically done using a delay-and-sum (DAS) method where a time delay, corresponding to time-of-flight, is calculated for each point in the image space, for each transmit-receive transducer pair. The received signal is then mapped over the entire image space for each receive transducer, with the resulting partial images summed across transducers to produce a final image. The key to DAS beamforming is that

the received echo data has no directional information, and therefore each transducer can only map echoes to an arc. However, if the received signals are summed over several transducers in different locations, the signals will only be coherent where the actual echo was produced.

Reconstruction algorithm.

The key computational step of DAS beamforming is the delay calculation given by

$$d_P = \frac{1}{c}\left(R_P + \sqrt{R_P^2 + x_i^2 - 2x_i R_P \sin\theta}\right) \tag{5.1}$$

where d_P is the round-trip delay from the center transducer to the point P to the receive transducer i, c is the speed of sound in tissue (1540 m/s), R_P is the radial distance of point P from the center of the transducer, θ is the angular distance of point P from the line normal to the center transducer, and x_i is the distance of transducer i from the center. As this calculation must be performed for every transmitter-receiver-point trio, it dominates the overall computation, especially for 3D systems where the number of transducers and points is expanded into the elevational dimension, producing over 1 billion unique delays per volume.

Imaging Task. The B-mode imaging task in MIRAQLE is representative of an abdominal 3D ultrasound scan to diagnose anechoic lesions in tissue (e.g., liver cysts). The radiologist's objective in this task is to clearly discern cysts from background tissue (e.g., to be able to precisely measure cyst dimensions). Inaccuracy in beamforming can cause noise (e.g., due to side and grating lobes) to appear within the cysts, which may make small cysts indistinct. Through simulation of ultrasonic wave propagation in Field II [41, 42] (a widely used MATLAB toolbox for simulating ultrasound physics), we have constructed radio frequency echo signals sampled at 40 MHz of a volume of tissue. The imaging volume contains 27 anechoic cysts in three, 3×3 grids ranging from 3-7 mm in diameter at depths ranging from 4-8 cm. Figure 5.1a

Table 5.1: Ultrasound task parameters.

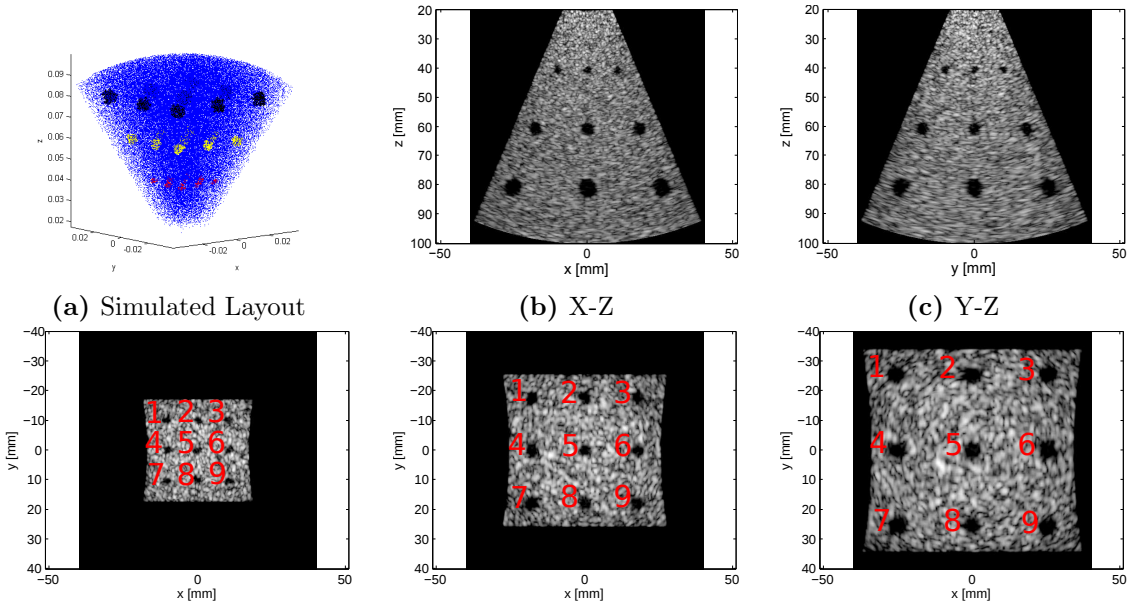
Parameter	Cyst imaging
Receive aperture size	120×88
Receive aperture pitch	$\frac{\lambda}{2}$
Transmit frequency	4 MHz
Sampling frequency (f_s)	40 MHz
Cyst depths	4-8 mm
Image resolution	96×96×4157
Image size	$\frac{\pi}{4}\times\frac{\pi}{4}\times 8$ cm
Transmits per volume	96

shows the layout of the cysts and tissue simulated. The benchmark reconstructs a 3D volumetric image from 96 transmissions using a 120×88 element (10,560 channel) transducer array. The resulting image is 96×96×4157 points covering a depth of 8 cm (2-10 cm) and lateral and elevational angles of $\pi/4$. Example 2D cross-sectional slices from reference beamformed volume are shown in Figure 5.1 with cyst positions labelled for each depth. Key parameters of the ultrasound system and imaging task scenario appear in Table 5.1.

Quality Metric. The key image quality metric is the distinguishability of the cysts from the surrounding tissue. The conventional metric used in ultrasound studies to quantify cyst imaging quality is the contrast-to-noise ratio (CNR), which is defined as:

$$\text{CNR} = \frac{|\mu_{\text{cyst}} - \mu_{\text{bgnd}}|}{\sqrt{\sigma_{\text{cyst}}^2 + \sigma_{\text{bgnd}}^2}} \tag{5.2}$$

where μ_{cyst} and μ_{bgnd} correspond to mean brightness of cyst and background, and σ_{cyst} and σ_{bgnd} are the standard deviation of cyst and background. The achievable CNR for a cyst varies with its diameter and depth. For optimizations that may introduce error, we recommend a conservative quality threshold where each cyst's CNR is at least 94.5% (0.5 dB degradation) of the CNR achieved in the unmodified reference run of the benchmark with double-precision floating point computations. Using an offline study we found that a variation in CNR of 94.5% is approximately one standard deviation from the mean or less for each cyst over varied random scatter



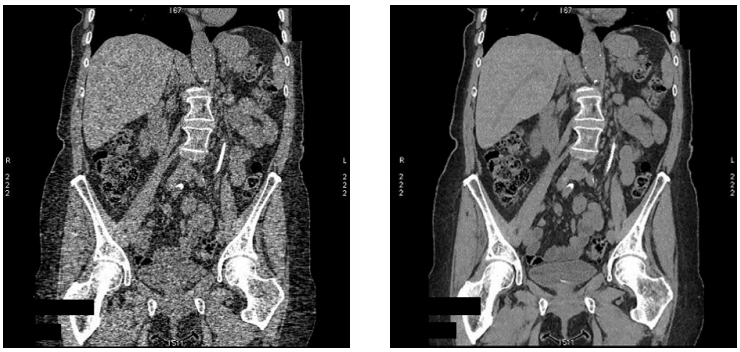
(a) Simulated Layout **(b)** X-Z **(c)** Y-Z

(d) X-Y, 4 cm **(e)** X-Y, 6 cm **(f)** X-Y, 8 cm

Figure 5.1: Ultrasonic cyst imaging reference; (a) Layout of simulated phantom data. Blue is the scatterers used for simulating tissue, red is the nine 3 mm diameter cysts located at 4 cm depth, yellow is the nine 5 mm diameter cysts located at 6 cm depth, and black is the nine 7 mm diameter cysts located at 8 cm. (b-f) Key 2D slices of reference beamformed image volume. Mid X-Z slice (b), mid Y-Z slice (c), and Y-Z slices are shown at depths 4 cm (d), 6 cm (e), and 8 cm (f) with numbered cyst positions shown for each depth.

distributions. As this amount of variation can occur naturally by simply having cysts located in different tissue or locations, any variation within this threshold should be acceptable for our analysis tool.

Architectural Opportunities. 3D ultrasonic beamforming presents several architectural challenges and opportunities. The raw data rate of the receive signal is extremely high, exceeding several terabits per second. There is a need for methods that can maintain image quality while compressing or discarding some of the input data to make the data rate more manageable. The trigonometric functions and square root in the delay calculation pose a significant computational burden. However, the calculation also admits massive parallelism, as each receive channel can be processed independently, and there is significant locality and regularity in the accesses to the receive signal. Optimizations that reduce computational burden or allow image quality



(a) **(b)**

Figure 5.2: Reconstruction via filtered back projection (a), compared to model-based iterative reconstruction (b) for a low-dose chest CT scan.

to be maintained with fewer transmissions per volume are clinically valuable because they allow an ultrasound system to achieve a higher frame rate (which is critical, for example, in cardiac imaging).

Unlike the other imaging modalities, ultrasound can be implemented in hand-held systems that consume only a few watts; hand-held 2D systems already in clinical use. In extending such systems to 3D, beamforming grows to dominate the power budget [74]. Hence, 3D ultrasound is highly sensitive to the energy efficiency of the image reconstruction.

5.1.2 X-Ray Computed Tomography

Background. In X-ray computed tomography (X-ray CT), X-ray data is collected over a series of angles with a fixed X-ray source and detector array set opposite each other, rotating around the patient. At each angle, the intensity of the signal received by each detector depends on the X-ray absorption of the ray through the patient from the source to that detector. The textbook approach to reconstruct an image from the collected data is called *filtered back projection*, or FBP. In this basic approach, the X-ray measurements are "back-projected" by assigning their absorption values to the voxels through which the corresponding ray passed, and voxel absorption values are combined across the angles. Furthermore, because the reversal process

of back-projecting is not a true mathematical inverse, a high boost filter (such as a ramp filter) is applied before the back-projection to prevent blurring. FBP has been the primary method for clinical CT reconstruction since the modality's inception.

Reconstruction algorithm. More recently, efforts have been made to reduce the X-ray dosage a patient receives during a scan by lowering the source intensity and/or the number of views (i.e., exposures). Unfortunately attempting to perform FBP on a low-dose scan results in poor image quality as shown in Figure 5.2a. However, using model-based iterative reconstruction (MBIR) [83] initialized FBP, a high quality image can be reconstructed (Figure 5.2b). The MBIR algorithm we include in MIRAQLE is similar to that approved by the FDA for clinical low-dose scans.

MBIR is a minimization problem that can be stated as:

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}\geq 0} \Psi(\boldsymbol{x}), \qquad \Psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2_{\boldsymbol{W}} + R(\boldsymbol{x}) \tag{5.3}$$

where $\boldsymbol{A}$ is the system matrix, $\boldsymbol{y}$ is the measurements, $\boldsymbol{W}$ is the statistical weighting and R is the regularizer. That is, the minimization seeks an image estimate $\widehat{\boldsymbol{x}}$ that minimizes a cost function comprising a data-fit term (which quantifies how well the estimate fits the measurements) and a regularization term (that reduces noise in the image).

The minimization problem is solved iteratively. Figure 5.3 illustrates the computational phases in each iteration, which are each separated by a global synchronization barrier. Each iteration, a gradient of Ψ is computed and used to update the current image result ($\boldsymbol{x}^{(n)}$). To estimate the gradient of Ψ, a "forward-projection" (computational emulation of the X-ray scan) is applied to $\boldsymbol{x}^{(n)}$ giving ($\boldsymbol{A}\boldsymbol{x}^{(n)}$). Next, the original received data is subtracted from the result ($\boldsymbol{A}\boldsymbol{x}^{(n)} - \boldsymbol{y}$). A statistical weighting ($\boldsymbol{W}$) is applied to this difference and then a back-projection ($\boldsymbol{A}^T$) is computed.

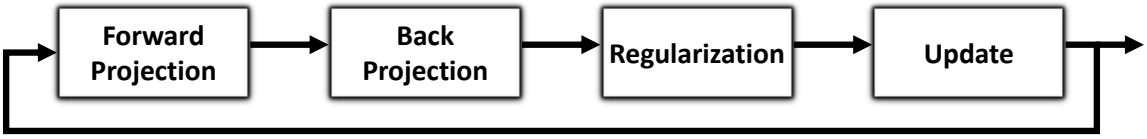


Figure 5.3: Visual representation of MBIR CT computation phases. Arrows represent global barriers between steps.

Finally an edge-preserving regularizer on $\boldsymbol{x}^{(n)}$ is computed to smooth out the image and added to the result, giving the final gradient estimation as

$$\nabla\Psi = \nabla\boldsymbol{R}(\boldsymbol{x}^{(n)}) + \boldsymbol{A}^T\boldsymbol{W}(\boldsymbol{A}\boldsymbol{x}^{(n)} - \boldsymbol{y}) \tag{5.4}$$

This result is multiplied with a diagonal majorizer at end of the iteration to produce the new image result, $\boldsymbol{x}$. The computation repeats for a specified number of iterations.

Imaging Task. MIRAQLE includes a MBIR CT imaging task representative of a low-dose helical chest CT scan. The benchmark task reconstructs a 512^3 voxel simulated scan of an XCAT phantom [79] using a detector with 888 channels, 64 rows and 8 helix turns of 984 views each, which are representative of a clinical scan. The XCAT phantom provides an accurate representation of complex human anatomy and is widely used in studies of X-ray CT algorithms. The resulting image should provide a noise-free view of anatomy with sharp edges and no visual artifacts.

Quality Metric. X-ray image intensities are reported on the Hounsfield scale that represents X-ray attenuation of materials. The attenuation of water is defined as 1000 Hounsfield Units (HU) and air is 0 HU. The displayed dynamic range for an anatomic image is typically about 400 HU (800-1200 HU). We quantify image quality as the root-mean-square (RMS) difference across all voxels between the output of the benchmark run and a reference reconstruction that we allow to run until the algorithm converges to the minimizer of the cost function. We set a threshold of 2 HU RMS error (0.5% of the dynamic range) for image quality to be acceptable.

We define the golden image for the MIRAQLE CT imaging task as the minimizer

to which the baseline iterative reconstruction algorithm ultimately converges (after a very large number of iterations). Note that this minimizer differs by a RMSE of about 10 HU from the known ground truth of the XCAT phantom due to geometric approximations in the X-ray system model. Closing this gap is beyond the scope of architecture research. However, architectural optimizations that introduce more than 2 HU RMS error from the baseline reference will likely produce artifacts that may obscure important anatomic detail.

Architectural Opportunities. MBIR CT is enormously computationally expensive, with clinical systems requiring 30-60 minutes of computation on a 112-core cluster to compute a single 3D image. The best way to parallelize MBIR CT algorithms remains an open research topic. In principle, each algorithmic step offers embarrassing parallelism over either the image or detector space, but efficiency can be improved by grouping calculations into threads that allow common sub-expressions to be factored out of inner-most loops and to maximize spatial locality and reuse within each thread. The implementation we provide in MIRAQLE parallelizes forward projection over views and back-projection and regularization over regions of the image space. Several aspects of the algorithm are amenable to SIMD parallelization, for example, across adjacent voxels. However, various complexities in memory access pattern arise that make both SIMD and SIMT parallelism difficult to exploit. As we show later, with sufficient thread parallelism, forward and back-projection become bandwidth-bound on modern servers, calling for architectural optimizations that conserve memory bandwidth.

5.1.3 Magnetic Resonance Imaging

Background. The basis of all magnetic resonance imaging (MRI) is the physical phenomenon of nuclear magnetic resonance, the process of electromagnetic excitation and relaxation of atomic nuclei in a magnetic field. As discussed in Chapter IV that

the nucleus of isotopes with an odd atomic number has an angular momentum, or spin, which in turn creates a nuclear magnetic moment, making such isotopes behave like magnetic dipoles and respond to magnetic fields. The spin of all magnetic dipoles is subjected to a strong magnetic field, inducing a steady-state precession around the axis of the primary field B_0. Then, by subjecting the nuclei to an excitation pulse B_1, a transverse oscillation is induced. When the excitation pulse is removed, the magnetization relaxes back to equilibrium. The relaxation time depends on the material's atomic composition, leading to different relaxation signals in different tissues. An image is reconstructed by localizing the relaxation signal by exciting the nuclei with a sequence of spatially dependent magnetic fields.

Reconstruction Algorithm. For conventional anatomical MRI, modern computer hardware can easily compute the Fourier transforms necessary to generate images. However, many modern dynamic MRI applications require large imaging volumes to be acquired at such high frame rates that complete Fourier sample information simply cannot be physically acquired fast enough. Hence, recent MRI research has turned to compressive sensing techniques with heavy computational burdens to create acceptable images, despite spatial under-sampling. MIRAQLE includes dynamic reconstruction via the low-rank + sparse-matrix decomposition (L+S) technique [25, 63]. The L+S method iterates between two key computational steps. First, it performs a non-uniform Fourier transform (NUFFT) [27, 80] on radially acquired spatial frequency (k-space) data into a Cartesian spatial grid. Second, it de-noises the image by decomposing the transformed signal matrix into a low-rank matrix (capturing slow-moving and static aspects of the image) and a sparse matrix (capturing fast movement and noise) and fitting each to the data separately.

For undersampled MRI data, the L+S reconstruction method is defined by the following optimization problem:

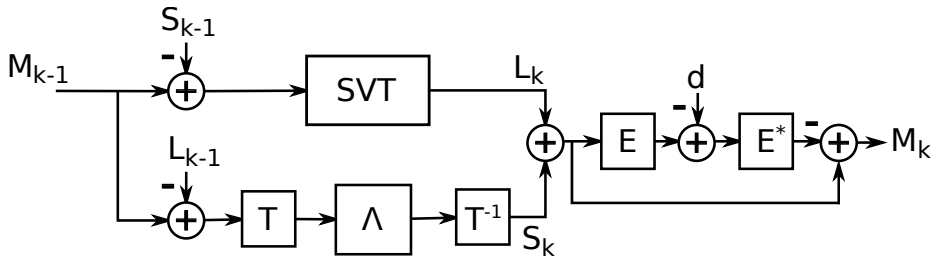


Figure 5.4: Single Iteration of L+S Algorithm. Reproduced from [63], the L+S algorithm of solving Eq. 5.5. In each iteration, $\boldsymbol{M}$ is decomposed into $\boldsymbol{L}$ and $\boldsymbol{S}$ which are independently updated. Afterwards, the values are combined and a final update is performed by removing the residual, creating the new value of $\boldsymbol{M}$.

$$\min_{L,S} \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{TS}\|_1 \quad \text{s.t.} \quad \boldsymbol{E}(\boldsymbol{L}+\boldsymbol{S}) = \boldsymbol{d} \tag{5.5}$$

where $\|\boldsymbol{L}\|_*$ is the nuclear norm of the slow-moving and static components $\boldsymbol{L}$, $\|\boldsymbol{TS}\|_1$ is the l_1-norm of the remaining dynamic information, $\boldsymbol{S}$, after applying a sparsifying transform $\boldsymbol{T}$, λ is tuning parameter for weighting $\boldsymbol{S}$, $\boldsymbol{E}$ is the encoding operator for the NUFFT and coil sensitivity maps, and $\boldsymbol{d}$ is the undersampled k-space data. At each step of the algorithm, the current image result, $\boldsymbol{M}_{k-1}$, is decomposed into $\boldsymbol{L}_{k-1}$ and $\boldsymbol{S}_{k-1}$, and then both are updated using singular value thresholding (SVT) on $\boldsymbol{L}_{k-1}$ and a soft thresholding in the $\boldsymbol{T}$-domain of $\boldsymbol{S}_{k-1}$. The update matrices $\boldsymbol{L}_k$ and $\boldsymbol{S}_k$ are recombined, and a residual is calculated using $\boldsymbol{d}$ which is subtracted to produce the updated image $\boldsymbol{M}_k$. Figure 5.4 illustrates the computation for a single iteration.

Imaging Task. MIRAQLE includes a dynamic contrast-enhanced (DCE) MRI reconstruction task of a 3D brain phantom based on the BrainWeb data [2, 15] and modified to have 10 mm radius tumor. The addition of the tumor and emulation of the contrast agent were done with the help of collaborator Mai Le. This data set provides the received signals for a sequence of 700 radial "spokes" of multiple 2D slices of the MRI brain scan. Due to the lengthy reconstruction runtimes, MIRAQLE reconstructs data as distinct 2D slices rather than as a full 3D image. However,

it is common to construct 3D MRI images in this manner by stacking separately reconstructed 2D slices.

Quality Metric. As with the X-ray CT imaging task, we define the quality metric for MRI as the RMS difference of the algorithm's output image relative to a golden output. Unlike X-ray CT, MRI does not produce output values on a scale with a straight-forward physical interpretation, so we define our error metric as in terms of a normalized RMS difference, which is unitless. We set a quality threshold of 0.1% NRMSD, below which the difference between two images is generally indistinguishable to the human eye.

Architectural Opportunities. An over-arching goal of MRI research is to maintain image quality with fewer field excitations (i.e., less input data), since relaxation time is physically limited and hence the number of excitations that can be performed in a given time interval is fixed. Faster acquisitions improve image quality, since patients cannot remain perfectly still and breathing motion blurs the image. The overall calculation time in the L+S method is dominated by the NUFFT calculations. NUFFT is amenable to hardware acceleration; a variety of prior studies have explored hardware acceleration of NUFFT in other contexts [47, 48]. Our case-study focuses on a particular acceleration opportunity: the interpolation operation that maps between the polar coordinates in which MRI data are acquired and the Cartesian grid of the image. This interpolation operation is similar to a 2D convolution over the k-space data and is amenable to hardware techniques that accelerate convolution [70].

5.2 Image Quality Case Studies

To demonstrate the utility of our benchmark suite and quality analysis tools, we perform case studies of a series of hypothetical architectural and software optimizations that introduce approximation error into the output image. Our objective is to

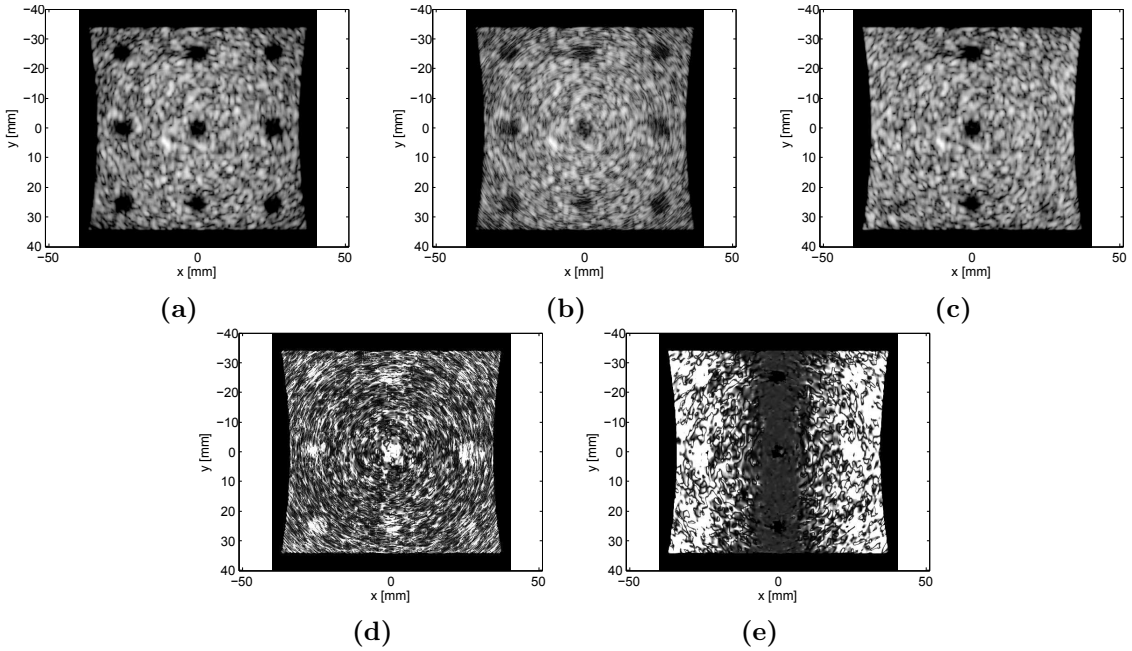
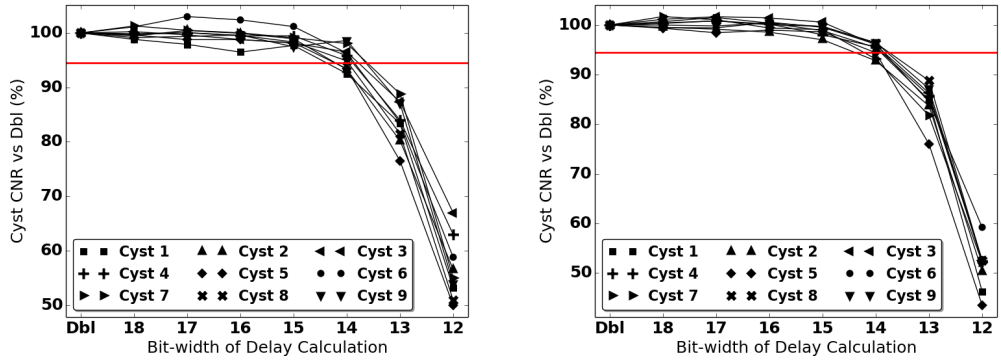


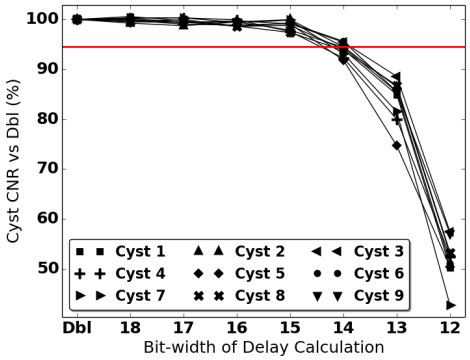
Figure 5.5: Ultrasound cyst imaging results on 40 dB dynamic range; 8 cm depth X-Y slice shown for: (a) MIRAQLE reference reconstruction with double-precision floating point and all receive channels. (b) Using 12-bit fixed point precision. Artifacts visible in all cysts. (c) Using a transducer step size of 4 (Fig. 5.8d). Cyst located at X = 0 are mostly unaffected due to elevational resolution being maintained; however, the lateral resolution is severely degraded with the rest of the cysts being nearly indistinguishable from the tissue. (d-e) Absolute difference of (b) and (c) with the reference. Images shown on a 0 to 5 dB scale.

demonstrate that MIRAQLE's quality metrics indicate when the approximation leads to visible degradation of the image.

For each imaging task, we first consider reducing the precision of a key computational step. Reducing precision is a canonical optimization for signal processing applications, as it can reduce memory bandwidth, increase effective cache capacity, and improve computational bandwidth and energy efficiency in special-purpose hardware pipelines. We also consider a task-specific optimization, which we describe in each case study below. Note that we model optimizations with simple changes to the benchmark source code (e.g., truncating precision after key computational steps); evaluating specialized hardware for each imaging modality is beyond the scope of this paper (indeed, we seek to enable such follow-on research).



(a) 4 cm **(b)** 6 cm



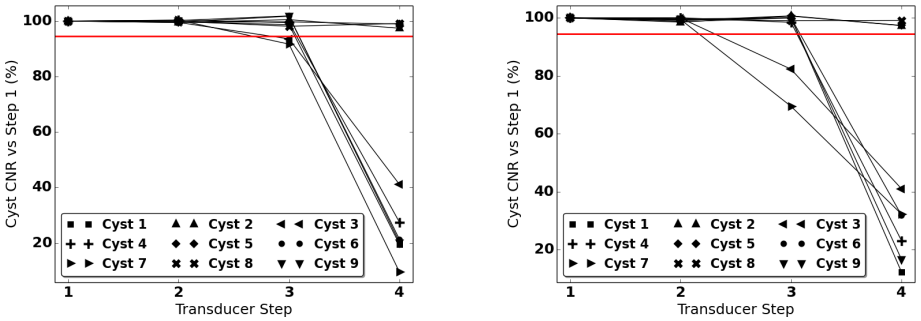
(c) 8 cm

Figure 5.6: CNR as a percentage of the reference CNR of the 27 cysts in the ultrasonic B-mode imaging task. "Dbl" represents the double-precision floating point reference. Image quality begins to degrade at 14 bits and substantially below 13 bits. Cysts at (a) 4 cm, (b) 6 cm, (c) 8 cm.

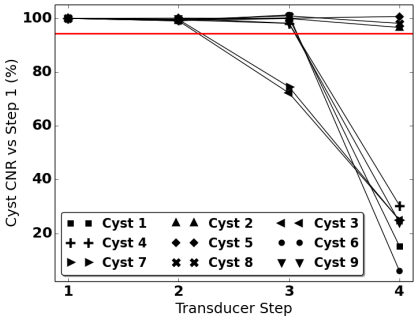
5.2.1 3D Ultrasound Imaging

Precision reduction. For the ultrasound B-mode imaging task, we consider the impact of reducing the precision of the delay calculation, using a fixed point rather than a floating point representation. Prior work on hardware-accelerated synthetic aperture beamforming [74] uses reduced-precision fixed-point arithmetic to improve energy efficiency. Hence, this case study is representative of ongoing research.

Figure 5.5a shows the 8 cm depth X-Y slice of the full-precision reference reconstruction, while Figure 5.5b shows the 12-bit reconstruction. Note that all of the cysts exhibit visible artifacts, regardless of position. This can also be seen in Fig. 5.5d



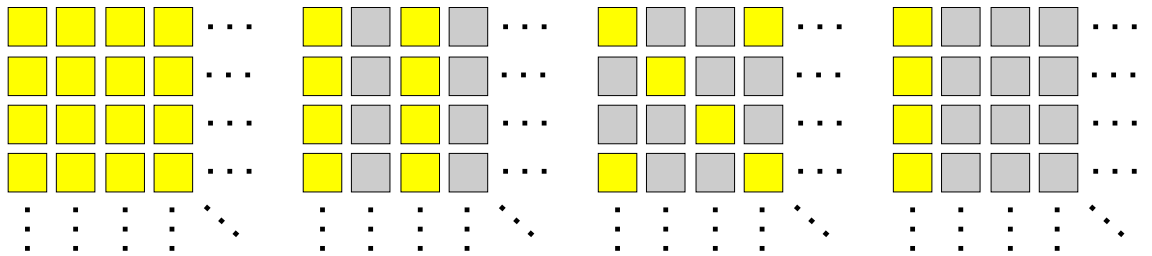
(a) 4 cm **(b)** 6 cm



(c) 8 cm

Figure 5.7: CNR variation due to step size as a percentage of the reference CNR of the 27 cysts in the ultrasonic B-mode imaging task. Step size of 1 represents the double-precision floating point reference. Cysts at (a) 4 cm, (b) 6 cm, (c) 8 cm. Image quality is maintained for step size of 2; however, cysts at positions 3 and 7 show strong degradation at all depths for step size 3. This is expected due to the diagonal-like pattern (shown in Fig. 5.8c). Step 4 shows nearly indistringuishable cysts except those along the X=0 plane (positions 2, 5, and 8). This again is due to the resulting transducer pattern (Fig. 5.8d) which is able to maintain strong elevational imaging quality.

which shows the absolute difference of the previously mentioned figures. We sweep the precision of the delay calculation from 12 to 18 bits and compare the CNR of the 27 cysts to a baseline reconstruction with double-precision floating point. Figure 5.6 shows the impact on CNR for each of the 27 cysts (higher is better). For the higher bit-widths the CNR is maintained with expected variation within our limits; however, at 14 bits, the CNR values begin to drop below the image quality requirement. As the bit width decreases further, we see that the CNR falls drastically with extreme degradation at 12 bits.



(a) Step Size 1 **(b)** Step Size 2 **(c)** Step Size 3 **(d)** Step Size 4

Figure 5.8: Transducer pattern resulting from varied step size. Yellow shows active transducers. (a) Default of step size 1, all transducers used. (b) Step size 2, due to the even width of the sub-aperture, this results in even columns being turned off. (c) Step size 3, with this step size not dividing the 32 wide sub-aperture evenly, a diagonal pattern emerges. (d) Step size 4, again an even multiple, resulting in three out of every four columns being turned off.

Channel reduction. Second, we consider an optimization that modifies the ultrasound receive aperture to reduce the number of receive channels processed in the reconstruction. We consider a reconstruction that skips some receive channels. It is common for ultrasound systems to receive with only a subset of channels to reduce data rates. This optimization is analogous to loop perforation or other architectural optimizations that selectively discard some input data.

We show the impact of discarding channels in Figure 5.7. Transducer step size refers to the distance between receive channels that are processed (i.e., 1 indicates that every channel is processed, 2 indicates every other, and so on) as shown in Fig. 5.8. Note that the reconstruction is quite tolerant of discarding a substantial number of receive channels. However, a step size of four results in catastrophic degradation of cysts not located in the X=0 plane (cyst positions 2, 5, and 8). Figures 5.5b and 5.5e show the corresponding image and absolute difference with the reference. Interestingly, cysts in the mid Y-Z plane have virtually no degradation from the reference and are well within our CNR limit despite the other cysts being hardly visible.

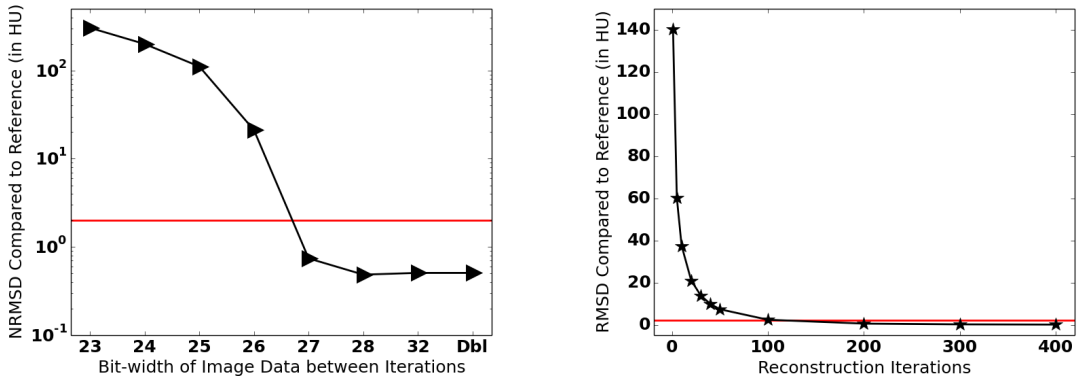


Figure 5.9: (a) RMSD of reconstructed CT image with reference using various fixed-point precisions for reconstructed image data between iterations (after back projection). "Dbl" uses a double precision float. (b) RMSD of reconstructed CT image with reference for various numbers of iterations.

5.2.2 Low-Dose X-Ray CT

Precision reduction. We investigate precision reduction in X-ray CT by truncating the image data to fixed-point precision during the back projection. Recall that back projection is the last step of the CT iteration before the regularizer is applied and is a primary component of the gradient calculation that produces the updates to the final image. Reducing the precision of the image drastically shrinks the memory and cache footprint of back projection and reduces memory bandwidth requirements. We compare the RMSD of the reconstructed image to our reference reconstruction with a tight quality bound of 2 HU (0.5%) error. Figure 5.9a shows that reduced precision has little impact for 27 bits and above; however, the RMSD increases well beyond the threshold once the width is reduced to 26 bits or fewer.

Additionally, we do not initialize using a FBP reconstruction for these simulations, causing large gradient values for early iterations. We believe this is the primary cause of the high bit-width requirement. Future investigations into changing the scale of the fixed-point across different iterations or starting with a FBP image may allow for further reduction beyond 27 bits.

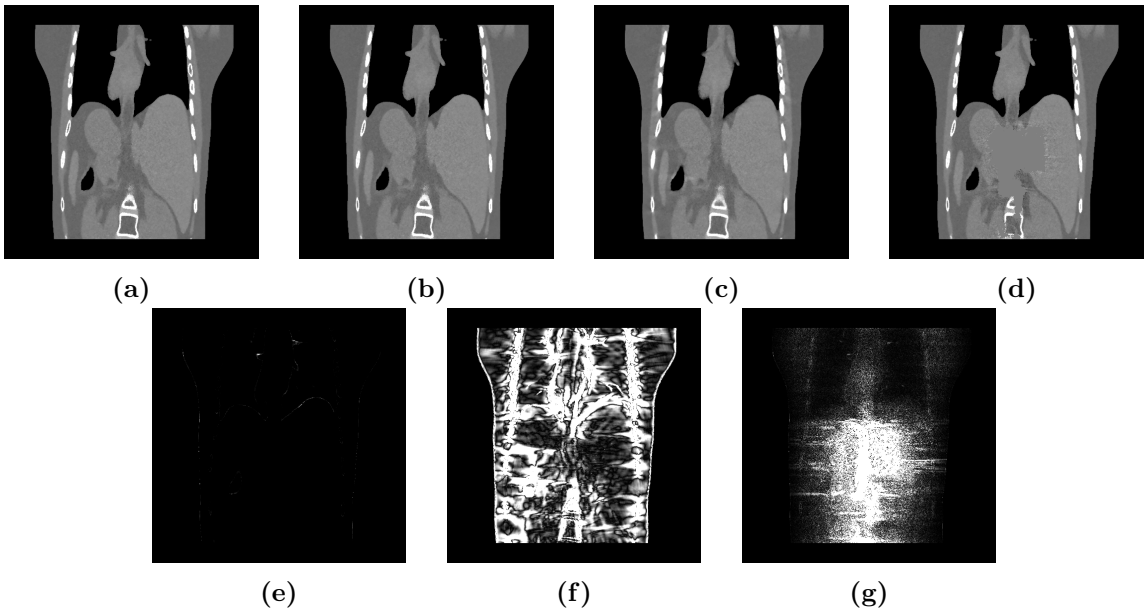
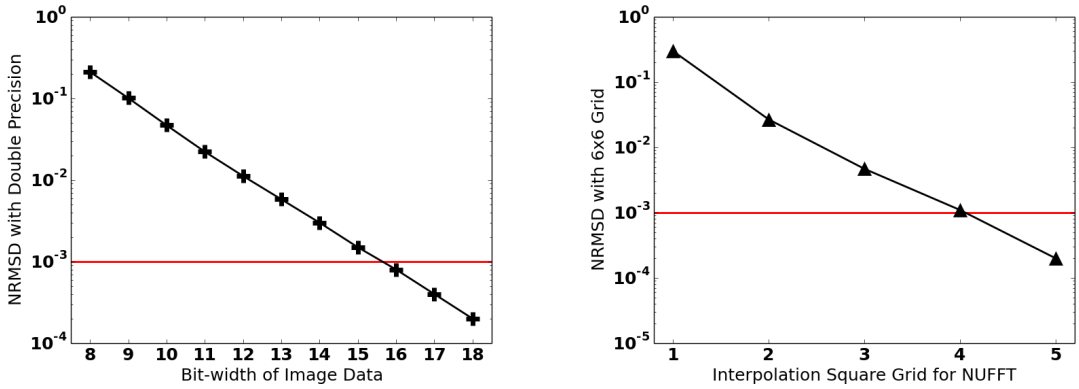


Figure 5.10: Comparison of Mid X-Z CT Slice on 800 to 1200 HU scale (a) Reference reconstruction of XCAT phantom [79]; (b) Reconstruction using 200 iterations and floating precision, indistinguishable from reference; (c) 10 iteration and floating precision, light artifacting over entire image with noticeable differences in the black region to the left of the spine and blurring around rib and spinal bones; (d) 200 iteration and 26-bit precision for back projection, severe artifacting in the middle with details indistinguishable in that region. (e-g) Absolute differences with reference for (b), (c), (d), respectively. Difference is shown on scale of 0-5 HU.

Early termination. Like all algorithms that use gradient descent to approach a minimizer, the accuracy of the CT reconstruction depends on the number of iterations for which the algorithm executes. Runtime can be trivially reduced by terminating early and outputting the image estimate that has been computed so far. Detecting convergence and deciding to stop the CT computation is challenging; approximation frameworks might try to adaptively predict when to stop.

In this study, we examine the impact on image quality when we vary the number of iterations. Figure 5.10 shows the results from ten and two hundred iterations as well as reference reconstruction, which establishes the ground truth for the benchmark. Two hundred iterations produces a result that is indistinguishable from the reference, while ten iterations exhibits an enlargement of the black area to the left of the spine as well as blurring of the visible rib and spinal bones. Figure 5.9b shows RMSD results



(a) (b)

Figure 5.11: (a) NMRSD of reconstructed MRI images using fixed-precision for image data (M, L, and S). "Dbl" uses double-precision float. (b) NMRSD of reconstructed MRI images with varied grid size for NUFFT interpolation.

for these and other values. One hundred iterations falls just short of our quality threshold, while two hundred is sufficient.

5.2.3 Dynamic MRI

Precision reduction. In dynamic MRI, we consider the impact of using a fixed-point representation of the image data and vary its precision. We quantize the low-rank (L), sparse (S), and full image (M) matrices during each iteration. Using a fixed-point representation reduces the storage requirement of these matrices, which can be exploited to improve cachability and communication/memory bandwidth. We vary precision from 8 to 18 bits and report our quality metric (normalized root-mean-square difference; NRMSD) relative to the reference reconstruction with double-precision floating point. Lower NRMSD is better, and we set a quality cut-off of 10^{-3} error for acceptable results. Figure 5.11a shows that the image quality degrades logarithmically with bit width; 16 bits of precision are needed to meet the cut-off. Fig. 5.12c shows a single frame of a 2D slice of the output image sequence for 8 bits, where visible degradation can be seen. In the degraded image a signficant drop in contrast occurs, particularly in the region between the brain and skull, and the

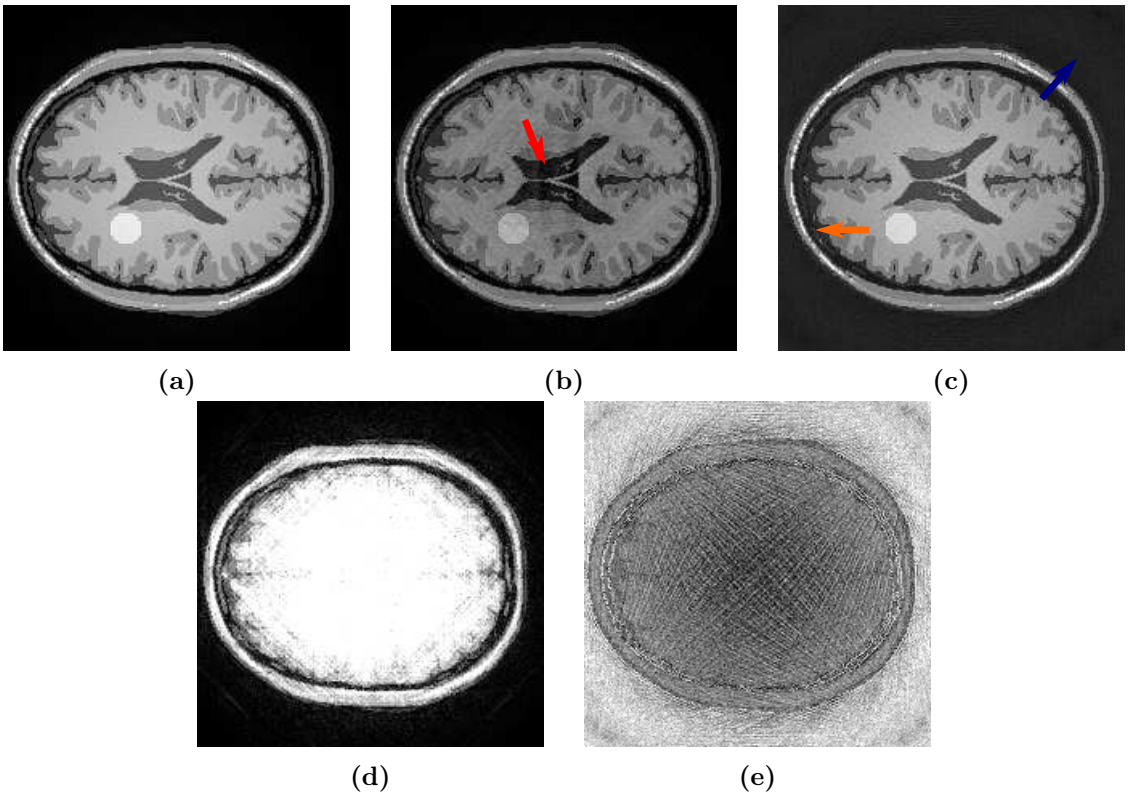


Figure 5.12: Comparison of Single Dynamic MRI Frame. Mid-slice of last reconstructed frame of 33 total output frames. Images are displayed on 0 to 13 scale. (a) Reconstruction performed with reference settings (6×6 interpolation, double precision). (b) Reconstruction using 1×1 interpolation. Artifacting is visible throughout the brain, particularly in the center (red arrow). (c) Reconstruction using 8-bit fixed point precision on L, S, and M image matrices. A severe loss of contrast is seen, particularly in the region between skull and brain (orange arrow) as well as outside the patient (blue arrow). (d-e) Absolute differences of (b) and (c) with reference shown on 0 to 2 scale.

outside of the patient has become visibly brighter. Fig. 5.12e highlights the areas of large differences, particularly the area outside the patient.

Reduced interpolation grid. Next, we consider the impact of reducing the interpolation grid used during the NUFFT computation, which accounts for a dominant portion of the algorithm runtime. The regridding operation constructs a Cartesian image space from the radial spoke data that comprises the input to the MRI reconstruction. A larger interpolation grid uses more input data (and correspondingly more computation) to calculate each image point. Evaluating a reduced interpolation

grid is highly relevant to hardware implementations of the NUFFT operation. The regridding is similar to a convolution operation and reducing the grid is analogous to shrinking the convolution kernel.

MIRAQLE's reference reconstruction uses a 6×6 interpolation grid. In this case study, we sweep the grid from 6×6 to 1×1. Figure 5.11b shows the image quality as the grid size is reduced as compared to the reference (6×6). A 5×5 grid is still sufficient to meet our target quality, but further reduction leads to unacceptable degradation, with 4×4 falling just outside our threshold. Figure 5.12b shows the selected output frame for 1×1 grid. The degradation shows that area outside the patient is maintained (unlike in the 8-bit case); however, a visible speckling is visible inside the brain tissue, and there is artifacting in the empty regions in the center. Additionally the overall image is significantly darkened overall compared to the reference despite using the same scale. Fig. 5.12d shows the absolute difference with the reference.

Despite the severe degradation of our test cases, the tumor is still clearly visible in both Fig. 5.12b and Fig. 5.12c. While this is partially due to the uniformity of the simulated brain tissue, it is entirely possible that such a tumor could be detected with similar errors even in a real-life imaging task. However, the tumor is not the focus of the task we have chosen, and such determinations are beyond the scope of this work. Instead we focus on maintaining overall image quality over all frames (including those where the contrast agent has not made the tumor visible). The tumor is instead used as a dynamic element that changes over temporal frames to ensure that reconstruction does not degrade even as parts of the image are changing (i.e., a high quality dynamic MRI is produced).

5.3 Bottlenecks & Opportunities

Finally, we discuss key computational bottlenecks that arise in the MIRAQLE imaging tasks when run on conventional server hardware, based both on our own ex-

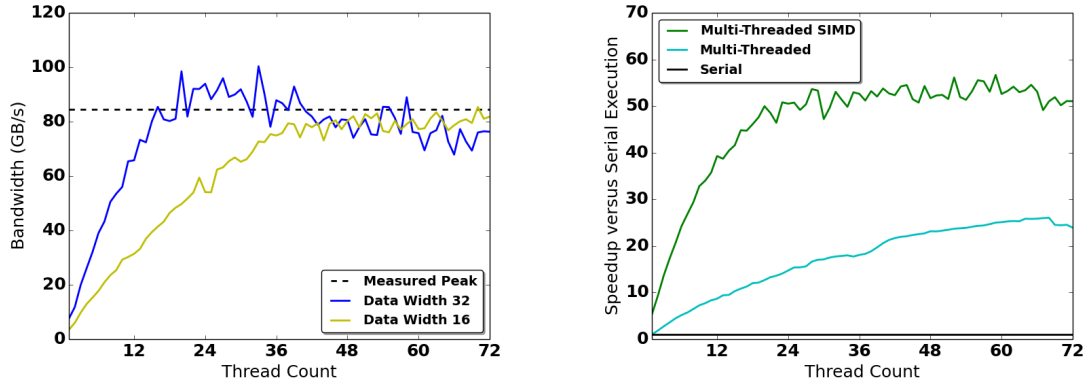


Figure 5.13: (a) Memory bandwidth usage during CT back projection for varied thread counts and 8-wide SIMD. System peak measured using STREAM triad benchmark [56, 57]. (b) Overall speed-up of CT reconstruction versus serial execution for multi-threaded and multi-threaded, 8-wide SIMD implementations.

periences and on results reported in the relevant literature from which we selected each task. Each of these bottlenecks represents an opportunity for algorithm-architecture co-design that might significantly accelerate image reconstruction.

5.3.1 Memory Bandwidth Constraints

Memory bandwidth poses a performance-limiting bottleneck in several of our imaging tasks. These bandwidth constraints ultimately limit the available thread-level parallelism in the tasks, as adding more threads simply leads to more memory stalls.

The memory bandwidth bottleneck is particularly acute in the X-ray CT reconstruction task. The CT reconstruction algorithm is easy parallelized across threads/cores, and initially scales nearly linearly with core count. However, both the forward and back projection steps (which dominate overall runtime) are extremely memory intensive; the number of computations per byte loaded from memory is small. With modern multi-core servers, it is easy to saturate available memory bandwidth well before all cores are fully utilized.

We illustrate the bandwidth constraint in the back projector in Figure 5.13a. Our back projector supports both 8-way SIMD and scalar execution on Intel hardware; here we consider the SIMD implementation. We vary the degree of thread parallelization from 1 to 72 threads on a 36-core (72-threaded) Intel Xeon server. The vertical axis shows the measured memory bandwidth consumption. The black line represents the peak bandwidth capability of the hardware, as measured by the STREAM triad benchmark [56, 57]. The blue line shows the bandwidth achieved by the CT back projector with SIMD enabled when the image is stored with single-precision floating point. Available memory bandwidth is saturated with only 16 threads. We also evaluated a reconstruction with half-precision (16-bit) floating point. Storing the image more densely doubles the effective memory bandwidth, and bandwidth saturates at roughly 32 threads.

Figure 5.13b shows how memory bandwidth constraint impacts performance. The blue line shows the performance of the CT back-projector (in terms of speedup relative to sequential execution) when SIMD support is disabled. Disabling SIMD alters the computation-to-memory-access ratio. As a result, even with 72 threads, memory bandwidth is not saturated and performance improves relatively steadily with increasing thread count. In contrast, performance of the 8-way SIMD implementation rapidly saturates around 20 threads. If memory bandwidth were not constrained, we estimate that the 8-way SIMD back projector could achieve nearly 150× improvement over sequential execution with 72 threads.

3D ultrasound faces similar memory bandwidth constraints. Due to the size of the transducer scanhead used to obtain the data, the received data must be transferred via cable for image reconstruction with data rate estimates as high as 6 Tb/s [74]. Commercial 3D designs like the Philips xMatrix probe attempt to overcome this limitation by performing a partial beamformation in the transducer head [30], and previous research [74, 75] has even proposed moving the entire beamforming computation into

the scan head to address the data transfer bottleneck.

These memory bandwidth constraints suggest further research into architectural mechanisms to use bandwidth more efficiently through precision reduction, compression, wider interfaces, or near-data processing.

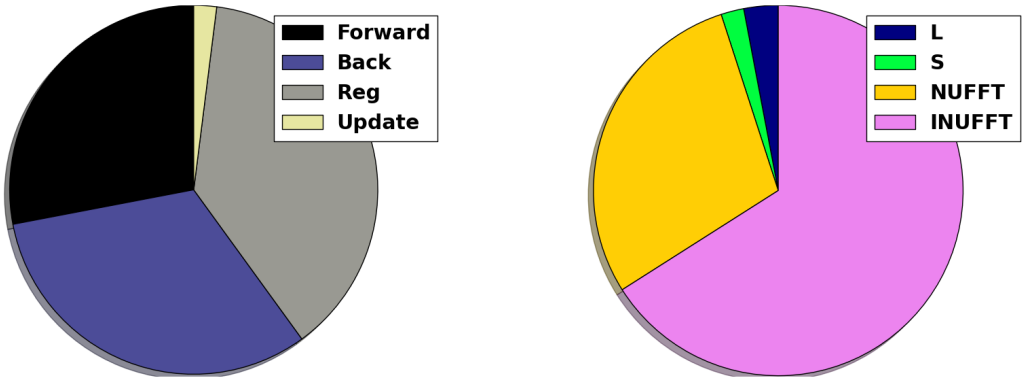
5.3.2 Unexploited Parallelism

Once memory bandwidth constraints are addressed, thread scalability constraints will pose the next key bottlenecks in several of the imaging tasks. For X-ray CT, prior work has shown that iterative reconstruction could be performed on a distributed system, achieving speed-up well into the hundreds of cores [71]. However, further scaling is hampered by communication bottlenecks and global barrier synchronization. Tighter integration of massively multicore systems could enable greater thread scalability.

Previous work on ultrasound [74] has shown that most of the computational steps of beamforming are embarrassingly parallel across channels, making the algorithm well-suited for highly parallel architectures. With channel counts of commercial designs well into the thousands [66], the ability to support many concurrent delay computations can result in near linear speed-up of the beamforming computation, enabling higher frame rates.

5.3.3 Optimized Memory Structures for Higher Dimensional Data

With X-ray CT iterating over 3D data and MRI iterating over 4D data, optimizing memory structures for higher dimensional structures is another area of particular interest. Much work has been done in developing specialized structures for 2D kernels [16, 70], but it may be possible to leverage newer technologies such as 3D stacking [8, 28] to create efficient memory accesses for these higher dimensional data structures. In particular, the regularization step of X-ray CT, which updates



(a) **(b)**

Figure 5.14: (a) Average runtime breakdown of CT iteration. A fairly equal time is spent among forward projection (28%), back projection (32%), and regularization steps(38%). (b) Average runtime breakdown for MRI iteration. The computation is heavily dominated by the NUFFT (29%) and inverse NUFFT (66%) of the update at the end of the computation (application of E and E^* in Figure 5.4).

a voxel based on its 26 neighbors in the 3D image, accounts for nearly 40% of the runtime (Fig. 5.14a). While these neighboring voxels are conceptually close the updating voxel, current 1D-based memory layouts cause them to be spread throughout memory, preventing dense data accesses. Specialized memory layout or hardware structures could be used to enable better 3D/4D locality, allowing for such neighbors to be accessed more quickly and efficiently.

5.3.4 Better use of SIMD/SIMT

Efforts to introduce SIMD into X-ray CT have met with mixed success [78] due to the challenging memory access patterns that require scatter/gather operations, which perform poorly on current SIMD implementations. At the same time, some algorithmic steps, such as forward projection, have inner loops with trip counts that vary for neighboring ray projections, which complicate SIMT implementation. Improving the scatter/gather capability of SIMD units could substantially improve the performance of several aspects of CT reconstruction.

5.3.5 Approximate Computing Techniques

Since medical image reconstruction algorithms must tolerate inherent analog sources of error, they can naturally tolerate error in the digital computation as well. Our case studies demonstrate the classic approach to exploit such error tolerance: precision reduction. However, myriad other approximate computing techniques are also potentially applicable to these image reconstruction applications, especially if portions of the algorithm are implemented in application-specific accelerators. Some authors have proposed aggressive voltage scaling beyond safe guardbands, which may lead to occasional timing errors in the data path that can be compensated through algorithmic techniques [21]. Others propose analog computation, which is inherently approximate [81]. Recent work has proposed imprecise adders [34, 85] and approximation in vector units [86]. Several other approximation approaches have been described in a recent survey paper [36].

5.3.6 Specialized Accelerators

A particularly promising architectural opportunity is the use of specialized hardware accelerators for the non-uniform FFT (NUFFT) computation. NUFFT and inverse NUFFT are performed at every iteration of the L+S algorithm (represented as $\boldsymbol{E}$ and $\boldsymbol{E}^*$ in Figure 5.4). Our timing analysis shows that these steps account for nearly 95% of the overall image reconstruction process, dominating the computation runtime (Fig. 5.14b. Prior research into developing specialized acceleration for NUFFT has shown great promise with speed-up and efficiency improvements over CPU and GPU implementations [10, 46]. However, the prior work did not target MRI applications and may need to be adapted to the unique matrix characteristics of the L+S algorithm.

5.4 Conclusions

We have introduced MIRAQLE, a new benchmark suite designed to bridge the gap between computer architects and medical imaging. MIRAQLE provides parallel, open-source implementations of state-of-the-art image reconstruction algorithms for several imaging modalities. There is enormous potential for architecture research to accelerate these algorithms and make them practical for clinical use. MIRAQLE's key feature is a set of image quality evaluation tools, designed in collaboration with experts in the selected modalities, to enable hardware-software co-design with quantitative evaluation of the impact of approximations on image quality. To demonstrate the capabilities of MIRAQLE, we perform case studies of algorithmic modifications that represent hypothetical hardware optimizations and show their impact on image quality. Finally we discuss key bottlenecks of each imaging algorithm and present opportunities for architectural innovation.

CHAPTER VI

Conclusions and Future Work

Medical imaging capabilities continue to improve, giving doctors safer and more reliable methods for evaluating patients. However, these improvements come with significant increases in computational burdens and are starting to push the limits of our current hardware. To enable the most state-of-the-art imaging techniques, we can no longer rely on Moore's Law to close the computational gap. Instead architects must focus their attention on the specialized support for these algorithms and utilize recent advancements in hardware design such as 3D die stacking, on-chip FPGAs, approximate computing, and specialized memory systems.

In this thesis we have discussed Sonic Millip3De, our custom hardware design for hand-held 3D ultrasound, to illustrate the benefits of increased hardware support for medical imaging. This design combines numerous architectural innovations with an algorithmic redesign to achieve orders-of-magnitude improvement in performance per watt, packing the capabilities of a large cart-based system into a small hand-held system. Additionally the reduced form-factor, 3D stacked design, and unique streaming design allow the hardware to overcome many of the issues that limit existing larger 3D systems such as memory bandwidth. By overcoming these limitations, Sonic Millip3De can achieve frame rates into the thousands, allowing it to enable 3D motion estimation techniques unsupported by current 3D ultrasound systems.

We have also discussed MIRAQLE, our new benchmark suite that bridges the gap between architects and medical imaging. MIRAQLE not only provides architects with the algorithms to enable better hardware design, it also provides them with image quality evaluation tools, enabling them to perform the same hardware-software co-design that was so critical in Sonic Millip3De's design. With these tools, we believe architects will be able to provide a more focused support for medical imaging and help make state-of-the-art imaging commonplace in the clinical setting.

6.1 Future Work

The key goal of this thesis has been to motivate and democratize medical image reconstruction research within the computer architecture community, and we hope that this work serves as the first step of many in the effort to develop better medical imaging hardware support. However, as this is only the beginning, there is still much work left to be done.

In Section 5.3 we outlined key areas of the medical imaging algorithms that are current bottlenecks or potential architectural opportunities. As discussed, future work in hardware development for medical image reconstruction should be broadly focused on improving memory accesses, exploiting greater parallelism. and developing specialized accelerators. In addition, we hope that architects will fully utilize the capabilities of our image quality evaluation tools to optimize the algorithm as well with the goal of achieving performance and efficiency gains similar to that of Sonic Millip3De. Some areas of particular interest with regard to algorithmic innovation are further development into disabling transducers in a sparse array pattern (extending the work in Section 5.2.1), variable scaling across iterations for fixed-width implementations (as discussed in Section 5.2.2), and specializing the NUFFT of MRI for specific sampling patterns.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] "Block Imaging - Medical Imaging Device Supplier," www.blockimaging.com.

[2] "Brainweb: Simulated brain database," http://www.brainweb.bic.mni.mcgill.ca/brainweb.

[3] "Here's What an Intel Broadwell Xeon with a Built-in FPGA Looks Like," www.theregister.co.uk/2016/03/14/intel_xeon_fpga.

[4] "TMS320C66x Multicore DSPs for High-performance Computing," http://www.ti.com/lit/ml/sprt619/sprt619.pdf.

[5] M. Ali, D. Magee, and U. Dasgupta, "Signal Processing Overview of Ultrasound Systems for Medical Imaging," Texas Instraments, Tech. Rep., Nov 2008.

[6] ARM, "Cortex-m3 40g specifications," http://www.arm.com/products/processors/cortex-m/cortex-m3.php.

[7] B. R. Benacerraf, C. B. Benson, A. Z. Abuhamad, J. A. Copel, J. S. Abramowicz, G. R. DeVore, P. M. Doubilet, W. Lee, A. S. Lev-Toaff, E. Merz *et al.*, "Three- and 4-dimensional ultrasound in obstetrics and gynecology proceedings of the american institute of ultrasound in medicine consensus conference," *Journal of ultrasound in medicine*, vol. 24, no. 12, pp. 1587–1597, 2005.

[8] B. Black, M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G. H. Loh, D. McCaule, P. Morrow, D. W. Nelson *et al.*, "Die stacking (3d) microarchitecture," in *Proc. of the 39th International Symp. on Microarchitecture*, 2006.

[9] D. Brasse, P. E. Kinahan, R. Clackdoyle, M. Defrise, C. Comtat, and D. W. Townsend, "Fast fully 3-d image reconstruction in pet using planograms," *IEEE transactions on medical imaging*, vol. 23, no. 4, pp. 413–425, 2004.

[10] U. I. Cheema, G. Nash, R. Ansari, and A. A. Khokhar, "Power-efficient regridding architecture for accelerating non-uniform fast fourier transform," in *Proc. of 24th International Conference on Field Programmable Logic and Applications (FPL '14)*, Sept 2014, pp. 1–6.

[11] A. Chien and V. Karamcheti, "Moore's law: The first ending and a new beginning," *Computer*, vol. 46, no. 12, pp. 48–53, 2013.

[12] Y.-k. Choi and J. Cong, "Acceleration of em-based 3d ct reconstruction using fpga," *IEEE transactions on biomedical circuits and systems*, vol. 10, no. 3, pp. 754–767, 2016.

[13] E. S. Chung, P. A. Milder, J. C. Hoe, and K. Mai, "Single-chip heterogeneous computing: Does the future include custom logic, FPGAs, and GPGPUs?" *Proc. of 43rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 225–236, Dec 2010.

[14] R. S. C. Cobbold, *Foundations of Biomedical Ultrasound*. Oxford University Press, 2007.

[15] D. L. Collins, A. P. Zijdenbos, V. Kollokian, J. G. Sled, N. J. Kabani, C. J. Holmes, and A. C. Evans, "Design and construction of a realistic digital brain phantom," *IEEE Transactions on Medical Imaging*, vol. 17, no. 3, pp. 463–468, June 1998.

[16] F. Conti and L. Benini, "A ultra-low-energy convolution engine for fast brain-inspired vision in multicore clusters," in *Proc. of the 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE '15)*. San Jose, CA, USA: EDA Consortium, 2015, pp. 683–688. [Online]. Available: http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=2755753.2755910

[17] J. Dahl, G. Trahey, and G. Pinton, "The effects of image degradation on ultrasound-guided hifu," in *Proc. of IEEE International Ultrasonics Symp.*, Oct. 2010, pp. 809–812.

[18] G. Dasika, K. Fan, and S. Mahlke, "Power-efficient medical image processing using PUMA," *Proc. of Symposium on Application Specific Processors (SASP '09)*, pp. 29–34, July 2009.

[19] G. Dasika, A. Sethia, V. Robby, T. Mudge, and S. Mahlke, "MEDICS: ultra-portable processing for medical image reconstruction," in *Proceedings of the 19th international conference on Parallel architectures and compilation techniques*. ACM, 2010, pp. 181–192.

[20] B. De Man and S. Basu, "Distance-driven projection and backprojection," in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, vol. 3, 2002, pp. 1477–80.

[21] Y. Emre and C. Chakrabarti, "Low energy motion estimation via selective aproximations," in *ASAP 2011-22nd IEEE International Conference on Application-specific Systems, Architectures and Processors*. IEEE, 2011, pp. 176–183.

[22] H. Erdogan and J. A. Fessler, "Ordered subsets algorithms for transmission tomography," *Phys. Med. Biol.*, vol. 44, no. 11, pp. 2835–51, Nov. 1999.

[23] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *Proc. of the 38th International Symp. on Computer Architecture*, 2011, pp. 365–376.

[24] H. Esmaeilzadeh, A. Sampson, L. Ceze, and D. Burger, "Architecture support for disciplined approximate programming," in *ACM SIGPLAN Notices*, vol. 47, no. 4. ACM, 2012, pp. 301–312.

[25] L. Feng, R. Grimm, K. T. Block, H. Chandarana, S. Kim, J. Xu, L. Axel, D. K. Sodickson, and R. Otazo, "Golden-angle radial sparse parallel MRI: Combination of compressed sensing, parallel imaging, and golden-angle radial sampling for fast and flexible dynamic volumetric MRI," *Mag. Res. Med.*, vol. 72, no. 3, pp. 707–17, Sep. 2014.

[26] J. A. Fessler, "Model-based image reconstruction for mri," *IEEE Signal Processing Magazine*, vol. 27, no. 4, pp. 81–89, 2010.

[27] J. A. Fessler and B. P. Sutton, "Nonuniform fast fourier transforms using min-max interpolation," *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 560–574, 2003.

[28] D. Fick, R. Dreslinski, B. Giridhar, G. Kim, S. Seo, M. Fojtik, S. Satpathy, Y. Lee, D. Kim, N. Liu *et al.*, "Centip3de: a 3930 dmips/w configurable near-threshold 3d stacked system with 64 arm cortex-m3 cores," in *Proc. of International Solid-State Circuits Conference*, Feb. 2012.

[29] Food and Drug Administration, "510(k) premarket notification submission for ge vec reconstruction option," 2011. [Online]. Available: http://www.accessdata.fda.gov/cdrh_docs/pdf10/K103489.pdf

[30] S. Freeman, "Microbeamforming for large-aperture ultrasound transducers," in *Proc. of the 53rd Annual Meeting of the American Association of Physicists in Medicine (AAPM '11)*, Aug. 2011.

[31] S. H. Fuller, L. I. Millett *et al.*, *The Future of Computing Performance: Game Over or Next Level?* National Academies Press, 2011.

[32] M. Goodsitt, H. Chan, E. Christodoulou, and S. Larson, "The effect of model based iterative reconstruction (GE-VEO) on the CT numbers and noise of both small lung nodules and large homogeneous (heart and spongiosa) regions in an anthropomorphic chest phantom," in *Proc. Amer. Assoc. Phys. Med.*, 2012, p. 4016.

[33] R. Gordon, R. Bender, and G. T. Herman, "Algebraic reconstruction techniques (art) for three-dimensional electron microscopy and x-ray photography," *Journal of Theoretical Biology*, vol. 29, no. 3, pp. 471 – 481, 1970.

[34] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "Impact: imprecise adders for low-power approximate computing," in *Proceedings of the 17th IEEE/ACM international symposium on Low-power electronics and design*. IEEE Press, 2011, pp. 409–414.

[35] P. A. Hager, A. Bartolini, and L. Benini, "Ekho: A 30.3 w, 10k-channel fully digital integrated 3-d beamformer for medical ultrasound imaging achieving 298m focal points per second," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 5, pp. 1936–1949, 2016.

[36] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *2013 18th IEEE European Test Symposium (ETS)*. IEEE, 2013, pp. 1–6.

[37] G. U. Haugen, K. Kristoffersen, and D. G. Wildes, "Ultrasound probe subaperture processing," Patent US7 527 592, 2009.

[38] M. Horowitz, E. Alon, D. Patil, S. Naffziger, R. Kumar, and K. Bernstein, "Scaling, power, and the future of cmos," in *IEEE InternationalElectron Devices Meeting, 2005. IEDM Technical Digest.* IEEE, 2005, pp. 7–pp.

[39] A. Ibrahim, P. 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 & Test in Europe Conference & Exhibition*. EDA Consortium, 2015, pp. 1683–1688.

[40] J. Jensen, M. B. Stuart, and J. A. Jensen, "Optimized plane wave imaging for fast and high-quality ultrasound imaging," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 63, no. 11, pp. 1922–1934, Nov 2016.

[41] J. Jensen, "Field: A program for simulating ultrasound systems," in *Nordicbaltic Conf. on Biomedical Imaging*, 1996, pp. 351–353.

[42] J. Jensen and N. 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, March 1992.

[43] K. Karadayi, C. Lee, and Y. Kim, "Software-based Ultrasound Beamforming on Multi-core DSPs," University of Washington, Tech. Rep., March 2011, http://www.ti.com/lit/wp/sprabo0/sprabo0.pdf.

[44] M. Karaman and M. O'Donnell, "Subaperture processing for ultrasonic imaging," *IEEE Trans. on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 45, no. 1, pp. 126–135, Jan. 1998.

[45] M. Karaman, P.-C. Li, and M. ODonnell, "Synthetic aperture imaging for small scale systems," *IEEE Trans. on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 42, no. 3, pp. 429–442, 1995.

[46] S. Kestur, S. Park, K. M. Irick, and V. Narayanan, "Accelerating the nonuniform fast fourier transform using fpgas," in *Proc. of 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM '10)*, May 2010, pp. 19–26.

[47] S. Kestur, K. Irick, S. Park, A. Al Maashri, V. Narayanan, and C. Chakrabarti, "An algorithm-architecture co-design framework for gridding reconstruction using fpgas," in *Proceedings of the 48th Design Automation Conference*. ACM, 2011, pp. 585–590.

[48] S. Kestur, S. Park, K. M. Irick, and V. Narayanan, "Accelerating the nonuniform fast fourier transform using fpgas," in *Field-Programmable Custom Computing Machines (FCCM), 2010 18th IEEE Annual International Symposium on*. IEEE, 2010, pp. 19–26.

[49] T. Kgil, S. D'Souza, A. Saidi, N. Binkert, R. Dreslinski, T. Mudge, S. Reinhardt, and K. Flautner, "Picoserver: using 3d stacking technology to enable a compact energy efficient chip multiprocessor," in *Proc. of the 12th Conf. on Arch. Support for Programming Languages and Operating Systems*, 2006.

[50] D. Kim, S. Ramani, and J. A. Fessler, "Combining ordered subsets and momentum for accelerated X-ray CT image reconstruction," *IEEE Trans. Med. Imag.*, vol. 34, no. 1, pp. 167–78, Jan. 2015.

[51] J. Liu, S. Venkataramani, S. V. Venkatakrishnan, Y. Pan, C. A. Bouman, and A. Raghunathan, "Embira: An accelerator for model-based iterative reconstruction," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 11, 2016.

[52] G. Lockwood, J. Talman, and S. Brunke, "Real-time 3-d ultrasound imaging using sparse synthetic aperture beamforming," *IEEE Trans. on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 45, no. 4, pp. 980–988, July 1998.

[53] G. H. Loh, "3d-stacked memory architectures for multi-core processors," in *Proc. of the 35th International Symp. on Computer Architecture*, 2008.

[54] D. P. Magee, "Iterative time delay values for ultrasound beamforming," Patent US 20 100 249 594, 2010.

[55] K. Malladi, F. Nothaft, K. Periyathambi, B. Lee, C. Kozyrakis, and M. Horowitz, "Towards energy-proportional datacenter memory with mobile dram," in *Proc. of 39th International Symp. on Computer Architecture*, June 2012, pp. 37–48.

[56] J. D. McCalpin, "Stream: Sustainable memory bandwidth in high performance computers," University of Virginia, Charlottesville, Virginia, Tech. Rep., 1991-2007, a continually updated technical report. http://www.cs.virginia.edu/stream/. [Online]. Available: http://www.cs.virginia.edu/stream/

[57] ——, "Memory bandwidth and machine balance in current high performance computers," *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter*, pp. 19–25, Dec. 1995.

[58] M. G. McGaffin and J. A. Fessler, "Alternating dual updates algorithm for x-ray ct reconstruction on the gpu," *IEEE transactions on computational imaging*, vol. 1, no. 3, pp. 186–199, 2015.

[59] B. Murmann, ""ADC Performance Survey 1997-2015"," http://www.stanford.edu/ murmann/adcsurvey.html.

[60] National Institutes of Health, "U01 FOA PAR-12-206 decreasing patient radiation dose from ct imaging: Achieving sub-msv studies," http://grants.nih.gov/grants/guide/pa-files/PAR-12-206.html, 2012.

[61] T. Nelson, J. Fowlkes, J. Abramowicz, and C. Church, "Ultrasound biosafety considerations for the practicing sonographer and sonologist," *J. of Ultrasound in Medicine*, vol. 28, no. 2, p. 139, 2010.

[62] O. Oralkan, A. S. Ergun, J. A. Johnson, M. Karaman, U. Demirci, K. Kaviani, T. H. Lee, and B. T. Khuri-Yakub, "Capacitive micromachined ultrasonic transducers: next-generation arrays for acoustic imaging?" *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 49, no. 11, pp. 1596–1610, Nov. 2002.

[63] R. Otazo, E. Candes, and D. K. Sodickson, "Low-rank plus sparse matrix decomposition for accelerated dynamic mri with separation of background and dynamic components," *Magnetic Resonance in Medicine*, vol. 73, no. 3, pp. 1125–1136, 2015. [Online]. Available: http://dx.doi.org/10.1002/mrm.25240

[64] R. Palmer, J. Poulton, B. Leibowitz, Y. Frans, S. Li, A. Fuller, J. Eyles, J. Wilson, M. Aleksic, T. Greer *et al.*, "A 4.3 gb/s mobile memory interface with power-efficient bandwidth scaling," in *Symp. on VLSI Circuits*, 2009.

[65] C. Passmann and H. Eermert, "A 100-mhz ultrasound imaging system for dermatologic and ophthalmologic diagnostics," *IEEE Trans. on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 43, no. 4, pp. 545–552, July 1996.

[66] Philips Healthcare, "X6-1 transducer array," http://www.usa.philips.com/healthcare/product/HC989605409281/x6-1-xmatrix-array.

[67] M. J. Pihl and J. A. Jensen, "A transverse oscillation approach for estimation of three-dimensional velocity vectors, part i: concept and simulation study," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 61, no. 10, pp. 1599–1607, Oct 2014.

[68] S. Z. Pinter, J. M. Rubin, O. D. Kripfgans, M. C. Treadwell, V. C. Romero, M. S. Richards, M. Zhang, A. L. Hall, and J. B. Fowlkes, "Three-dimenional ultrasound measurement of blood volume flow in the umbilical cord," *Journal of Ultrasound in Medicine: Official Journal of the American Institute of Ultrasound in Medicine*, vol. 31, no. 12, pp. 1927–1934, 2012.

[69] J. L. Prince and J. M. Links, *Medical Imaging: Signals and Systems*. Pearson Prentice Hall, 2006.

[70] W. Qadeer, R. Hameed, O. Shacham, P. Venkatesan, C. Kozyrakis, and M. A. Horowitz, "Convolution engine: Balancing efficiency & flexibility in specialized computing," in *Proceedings of the 40th Annual International Symposium on Computer Architecture (ISCA '13)*. New York, NY, USA: ACM, 2013, pp. 24–35. [Online]. Available: http://doi.acm.org/10.1145/2485922.2485925

[71] J. M. Rosen, J. Wu, T. F. Wenisch, and J. A. Fessler, "Iterative helical CT reconstruction in the cloud for ten dollars in five minutes," in *Proc. of International Meeting on Fully 3D Image Reconstruction in Radiology and Nuclear Medicine*, 2013, pp. 241–244.

[72] J. M. Rubin, M. Feng, S. W. Hadley, J. B. Fowlkes, and J. D. Hamilton, "Potential use of ultrasound speckle tracking for motion management during radiotherapy: preliminary report," *J Ultrasound Med*, vol. 31, no. 3, pp. 469–481, Mar 2012.

[73] J. M. Rubin, J. C. Horowitz, T. H. Sisson, K. Kim, L. A. Ortiz, and J. D. Hamilton, "Ultrasound strain measurements for evaluating local pulmonary ventilation," *Proc. of IEEE International Ultrasonics Symposium (IUS '15)*, pp. 1–5, Oct 2015.

[74] R. Sampson, M. Yang, S. Wei, C. Chakrabarti, and T. F. Wenisch, "Sonic Millip3De: Massively parallel 3D stacked accelerator for 3D ultrasound," in *19th IEEE International Symposium on High Performance Computer Architecture*, Feb. 2013, pp. 318–329.

[75] ——, "Sonic Millip3De with dynamic receive focusing and apodization optimization," in *Proc. of the 2013 IEEE International Ultrasonics Symposium (IUS '13)*, July 2013, pp. 557–560.

[76] ——, "Sonic Millip3De: An architecture for handheld 3D ultrasound," *IEEE MICRO Top Picks in Computer Architecture of 2014 (Top Picks '14)*, May/June 2014.

[77] R. Sampson, M. Yang, S. Wei, R. Jintamethasawat, B. Fowlkes, O. Kripfgans, C. Chakrabarti, and T. F. Wenisch, "FPGA implementation of low-power 3D ultrasound beamformer," *Proc. of IEEE International Ultrasonics Symposium (IUS '15)*, Oct. 2015.

[78] R. Sampson, M. G. McGaffin, T. F. Wenisch, and J. A. Fessler, "Investigating multi-threaded SIMD for helical CT reconstruction on a CPU," *To appear at 4th International Conference on Image Formation in X-Ray Computed Tomography*, July 2016.

[79] W. P. Segars, G. Sturgeon, S. Mendonca, J. Grimes, and B. M. W. Tsui, "4d xcat phantom for multimodality imaging research," *Medical Physics*, vol. 37, no. 9, pp. 4902–4915, 2010. [Online]. Available: http://scitation.aip.org/content/aapm/journal/medphys/37/9/10.1118/1.3480985

[80] L. Sha, H. Guo, and A. W. Song, "An improved gridding method for spiral mri using nonuniform fast fourier transform," *Journal of Magnetic Resonance*, vol. 162, no. 2, pp. 250–258, 2003.

[81] R. St Amant, A. Yazdanbakhsh, J. Park, B. Thwaites, H. Esmaeilzadeh, A. Hassibi, L. Ceze, and D. Burger, "General-purpose code acceleration with limited-precision analog computation," *ACM SIGARCH Computer Architecture News*, vol. 42, no. 3, pp. 505–516, 2014.

[82] S. Stergiopoulos, Ed., *Advanced Signal Processing: Theory and Implementation for Sonar, Radar, and Non-Invasive Medical Diagnostic Systems*, ser. The Electrical Engineering and Applied Signal Processing Series. CRC Press, 2009.

[83] J.-B. Thibault, K. D. Sauer, C. A. Bouman, and J. Hsieh, "A three-dimensional statistical approach to improved image quality for multislice helical ct," *Medical Physics*, vol. 34, no. 11, pp. 4526–4544, 2007.

[84] K. F. Üstüner and G. L. Holley, "Ultrasound imaging system performance assessment," American Assocation of Physicists in Medicine Annu. Meeting, 2003.

[85] M. Vasudevan and C. Chakrabarti, "Image processing using approximate datapath units," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2014, pp. 1544–1547.

[86] S. Venkataramani, V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan, "Quality programmable vector processors for approximate computing," in *Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture*. ACM, 2013, pp. 1–12.

[87] B. Verbruggen, M. Iriguchi, and J. Craninckx, "A 1.7mw 11b 250ms/s 2x interleaved fully dynamic pipelined sar adc in 40nm digital cmos," in *Proc. of 2012 IEEE International Solid-State Circuits Conf.*, Feb. 2012.

[88] G.-M. von Reutern, M.-W. Goertler, N. M. Bornstein, M. D. Sette, D. H. Evans, M.-W. Goertler, A. Hetzel, M. Kaps, F. Perren, A. Razumovky, T. Shiogai, E. Titianova, P. Traubner, N. Venketasubramanian, L. K. Wong, and M. Yasaka, "Grading carotid stenosis using ultrasonic methods," *Stroke*, vol. 43, no. 3, pp. 916–921, 2012. [Online]. Available: http://stroke.ahajournals.org/content/43/3/916.abstract

[89] S. Wei, M. Yang, C. Chakrabarti, R. Sampson, T. F. Wenisch, O. Kripfgans, and J. B. Fowlkes, "A low complexity scheme for accurate 3d velocity estimation in ultrasound systems," in *2014 IEEE Workshop on Signal Processing Systems (SiPS)*. IEEE, 2014, pp. 1–6.

[90] S. Winkelmann, T. Schaeffter, T. Koehler, H. Eggers, and O. Doessel, "An optimal radial profile order based on the golden ratio for time-resolved MRI," *IEEE Trans. Med. Imag.*, vol. 26, no. 1, pp. 68–76, Jan. 2007.

[91] G. A. Wright, "Magnetic resonance imaging," *IEEE Signal Processing Magazine*, vol. 14, no. 1, pp. 56–66, 1997.

[92] M. Wu and J. A. Fessler, "GPU acceleration of 3D forward and backward projection using separable footprints for X-ray CT image reconstruction," in *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med*, 2011, pp. 56–9.

[93] M. Yang, R. Sampson, S. Wei, T. F. Wenisch, and C. Chakrabarti, "High frame rate 3-D ultrasound imaging using separable beamforming," *Journal of Signal Processing Systems*, vol. 78, no. 1, pp. 73–84, Jan. 2014.

[94] ——, "High volume rate, high resolution 3D plane wave imaging," *Proc. of IEEE International Ultrasonics Symposium (IUS '14)*, Sept. 2014.

[95] ——, "Separable beamforming for 3-D medical ultrasound imaging," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 279–290, Jan. 2015.

[96] M. Yang, R. Sampson, T. F. Wenisch, and C. Chakrabarti, "Separable beamforming for 3-D synthetic aperture ultrasound imaging," *Proc. of IEEE Workshop on Signal Processing (SiPS '13)*, Oct. 2013.

[97] M. Yang, "In support of high quality 3-D ultrasound imaging for hand-held devices," Ph.D. dissertation, Arizona State University, 2014.

[98] F. Zhang, A. Bilas, A. Dhanantwari, K. N. Plataniotis, R. Abiprojo, and S. Stergiopoulos, "Parallelization and performance of 3d ultrasound imaging beamforming algorithms on modern clusters," in *Proc. of the International Conf. on Supercomputing*, 2002.