Wireless, Real-Time Plane-Wave Coherent Compounding on an iPhone: A Feasibility Study

Cameron Lowell Palmer, *Student Member IEEE*, and Ole Marius Hoel Rindal, *Student Member, IEEE*

***Abstract*— The processing power in commercially available hand-held devices has improved dramatically in recent years. In parallel, techniques used in high-frame-rate medical ultrasound imaging, especially plane-wave (PW) imaging, have reduced the number of ultrasound transmissions and amount of data necessary to reconstruct an ultrasound image. In combination, the processing power and data reduction allow all of the processing steps in ultrasound image formation, from raw ultrasound channel data to final rendering, to be performed on a hand-held device. In this study, we send the raw ultrasound channel data from a research scanner wirelessly to an *off-the-shelf* hand-held device. The hand-held unit's graphical processing unit is processing the raw ultrasound data into the final image, achieving real-time frame rates on the order of 60–90 frames per second (FPS) for a single-angle PW transmission. Higher quality images are achieved by trading off frame rate by coherently compounding multiple PW images, resulting in frame rates on the order of, e.g., 13 FPS when coherently compounding 7 PW transmissions. The presented setup has the potential of providing image quality which could be valuable for simple medical ultrasound diagnostic scans of, e.g., the carotid artery or thyroid. Also, since the computationally expensive beamforming can be done in *off-the-shelf* devices, this could reduce the price of hand-held ultrasound systems in the future.**

***Index Terms*— Coherent plane-wave (PW) compounding, hand-held ultrasound, software beamforming, wireless probe.**

I. Introduction

HAND-HELD ultrasound systems are receiving increased attention and use [1], [2], and Godt [3] has predicted hand-held ultrasound will even replace the physician's stethoscope. However, ultrasound signal processing and image creation are computationally expensive. Until recently, most clinical high-end ultrasound scanners used specialized hardware to do ultrasound image formation. Ultrasound image formation, known as beamforming, is done by converting and combining the ultrasound signals received on the numerous elements inside the ultrasound probe into a brightness value in the image. Only in recent years has computing power increased to the point that manufacturers of high-end clinical ultrasound scanners could move beamforming into software [4], [5]. Software beamforming allows much more flexible implementations of beamforming algorithms potentially leading to improved image quality [6]–[11] by allowing real-time implementations of sophisticated beamforming techniques [12]–[15]. It is, therefore, comprehensible that most research ultrasound systems utilize software beamforming [16].

The benefit of a hand-held system is the ability for a physician to carry an ultrasound system in their pocket. Perhaps, more importantly, is the availability of medical imaging *on the scene* [17]. A paramedic can easily bring a device to the scene of an accident and within seconds perform for example a Focused Assessment with Sonography in Trauma (FAST) exam, allowing prearrival notification to the hospital, in turn, allowing a specialized team to be ready in the operating room. In addition, ultrasound images can be transmitted real time from the scene or ambulance, allowing physicians to review and assist the paramedics in the treatment of the patients before they even arrive at the hospital.

In the range of hand-held medical ultrasound devices, specialized hardware implementations of the beamforming algorithms are used in most commercially available devices [18]–[22]. In systems with wireless probes, the bottleneck of transferring large amounts of raw channel data led to all, or parts, of the beamforming to be implemented in dedicated hardware in the ultrasound probe. One technique for such processing is the synthetic aperture sequential beamforming (SASB) [23]. SASB has been demonstrated in implementations for commercially available hand-held devices [24], and the SASB approach motivated a system-level design of a wireless ultrasound probe [25]. The SASB uses focused transmission and performs the first stage of the beamforming in the probe, reducing the amount of data needed to be transferred to the probe significantly.

Recently, the miniaturization of high-performance processing hardware such as graphical processing units (GPUs) has made such high-end hardware available on *off-the-shelf* hand-held devices such as an iPad or iPhone. The hardware

Manuscript received January 28, 2019; accepted April 26, 2019. Date of publication May 2, 2019; date of current version July 10, 2019. This work was supported in part by the Center for Innovative Ultrasound Solutions (CIUS) and the Research Council of Norway under Project 237887. *(Cameron Lowell Palmer and Ole Marius Hoel Rindal contributed equally to this work.) (Corresponding author: Ole Marius Hoel Rindal.)*

C. L. Palmer is with the Department of Circulation and Medical Imaging, Faculty of Medicine, Norwegian University of Science and Technology, 7491 Trondheim, Norway (e-mail: cameron.palmer@ntnu.no).

O. M. H. Rindal is with the Research Group for Digital Signal Processing and Image Analysis, Department of Informatics, University of Oslo, 0315 Oslo, Norway (e-mail: omrindal@ifi.uio.no).

Digital Object Identifier 10.1109/TUFFC.2019.2914555

improvements have motivated software implementations of other computationally expensive parts of the ultrasound processing chain such as vector flow imaging on commercial, consumer-level tablets [26], [27].

Another recent breakthrough in medical ultrasound imaging is ultrafast imaging [28]. This was made possible through the introduction of plane-wave (PW) imaging [29]. In PW imaging, one can form the complete ultrasound image from the transmission of a single plane wave, compared to the conventional focused transmission that requires hundreds of transmissions per image. This dramatically reduced the amount of channel data necessary to create an image. Higher image quality can be obtained by coherently compounding multiple low-quality plan-wave images transmitted at different angles. The drawback of PW imaging is the reduced penetration depth and fewer tissue harmonics, making harmonic imaging a challenging task.

Fourier beamforming [30], [31] has emerged as a technique that can reduce the computational load of plane wave coherent compounded beamforming compared to conventional delay-and-sum beamforming [32]. Thus, a Fourier beamforming approach could possibly lower the computational load on a hand-held device even more. In particular, since most hand-held devices have specialized implementations, and potentially hardware, for the fast Fourier Transform algorithm [33]. Beamforming in the Fourier domain has also been suggested as a path to do compressed sensing and reducing the amount of data needed to reconstruct the full image [34]. Even though these studies and techniques are interesting and promising for a hand-held device, we limited our study to conventional time-delayed beamforming.

In this study, we will exploit the benefits from coherent PW compounded (CPWC) imaging and the increased computer power available on commercially available hand-held devices. We will demonstrate that it is possible to do real-time ultrasound image formation on an iPhone/iPad exploiting the processing power of the built-in GPUs. Contrary to the SASB technique, which used focused transmission and two-stage beamforming to reduce the amount of data needed to be transmitted, we use CPWC to reduce the amount of channel data needed per image, which also provides a tradeoff between the amount of data to be transferred and image quality. The current study builds on our previous study [35] and a related study [36]. Compared to our previous study, we have introduced CPWC imaging and expanding aperture imaging resulting in higher image quality, and several improvements to the processing chain resulting in dramatic improvements to the processing frame rate.

Our primary hypothesis is that an *off-the-shelf* hand-held device, such as an iPhone, has the processing power to do real-time ultrasound image formation from raw channel data. Here, the channel data are defined as the signals from the individual elements in an ultrasound probe. Second, the transmission of channel data from the probe to the iPhone can be done wirelessly, which suggests that a full ultrasound imaging system only needs a wireless probe with minimal processing of the signals, while an iPhone does the computationally expensive image reconstruction. This is different from the existing solutions which all use specialized hardware to do the beamforming [18]–[22]. The wireless probe needs the electronics and power to transmit and receive ultrasonic signals in the individual probe elements, and needs to be able to sufficiently sample, temporarily store, and transmit the raw channel data to the phone. This can be done in specialized hardware or field-programmable gate arrays, while the full beamforming processing chain can be processed by flexible software implementations in *off-the-shelf* hand-held devices. By having a software implementation of the beamforming system, progress in beamforming techniques can easily be incorporated, and improvements in the processing power of commercially hand-held devices can be exploited via software updates while still receiving the same type of data from the probe. We thus believe that future commercial devices will move more toward using *off-the-shelf* devices for much of the processing to lower the production cost and eventually device cost.

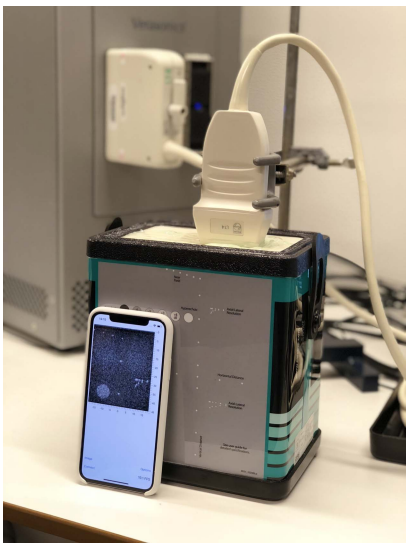


Fig. 1. Complete setup of the ultrasound probe imaging a phantom with the research scanner streaming the channel data wirelessly through the router before it is received, beamformed, and displayed on the iPhone. A demo video is available at https://youtu.be/oN8cwysGxyM.

In Section II, we detail the specific hardware used in our setup and detail the processing pipeline and choices made in the implementation and the networking details. The results are presented in Section III and discussed in Section IV. We conclude in Section V that we are capable of doing real-time ultrasound image formation on an iPhone processing multiple coherently compounding plane waves recorded from an ultrasound probe, as illustrated in Fig. 1.

II. Materials and Methods

A. Wireless Probe and Channel Data Acquisition

To mimic a wireless probe, we use a research scanner streaming raw recorded channel data wirelessly to the iPhone using a Unifi (Ubiquiti Networks, Inc., New York, NY, USA) ac-HD-Pro access point. We are using a Verasonics Vantage 256 (Verasonics Inc., Redmond, WA, USA) scanner with an L7-4 (Philips, Amsterdam, The Netherlands) linear probe with 128 elements with λ spacing (0.298 mm). The transmission sequence is a varying number of plane waves

TABLE I
MAIN TECHNICAL SPECIFICATIONS OF THE DEVICES USED IN THIS STUDY. *THE THREE LAST COLUMNS ARE BENCHMARK RESULTS FROM HTTPS://BROWSER.GEEKBENCH.COM/IOS-BENCHMARKS

Device	CPU	GPU	Memory	Wi-Fi	Single threaded*	Multi threaded*	Metal*
iPhone 6s+	A9, 2-core @ 1849 MHz	PowerVR GT7600 (six-core)	2 GB	802.11ac	2409	4161	10355
iPhone 8	A11 Bionic, 6-core @ 2390 MHz (2x Monsoon + 4x Mistral)	Apple GPU (three-core graphics)	2 GB	802.11ac MIMO	4225	10145	15137
iPhone X	A11 Bionic, 6-core @ 2390 MHz (2x Monsoon + 4x Mistral)	Apple GPU (three-core graphics)	3 GB	802.11ac MIMO	4213	10148	15239
iPad Pro 2017	A10X Fusion, 3-core @ 2340 MHz	PowerVR Series 7XT (12-core)	4 GB	802.11ac HT80 MIMO	3915	9340	29531

transmitted with a center frequency of 5.1 MHz with maximum and minimum angles at $\alpha = \pm 16°$, except for the case of a single plane wave which is transmitted at $\alpha = 0°$. The data are in-phase and quadrature sampled (IQ-sampled) using the bandwidth sampling data acquisitions available with the Verasonics Vantage systems [37]. In our setup, the number of samples in the IQ-sampled signal can be reduced by an integer factor, specifically 4, compared to the default RF-sampling frequency. However, the IQ-samples are complex, thus having both a real and an imaginary part so effectively the number of samples in the channel data is reduced by a factor of 2 using IQ-sampling. Since the signal is IQ-sampled, the envelope can be detected by simply taking the magnitude of the signal, avoiding the Hilbert transform needed to detect the envelope from an RF-signal. We are imaging a 45.8-mm-deep image requiring us to receive 512 samples per channel. The total amount of data (in bytes) per image frame is then

$$\frac{\text{bytes}}{\text{img}} = \frac{\text{PWs}}{\text{img}} \cdot \frac{\text{channels}}{\text{img}} \cdot \frac{\text{samples}}{\text{channel}} \cdot \frac{\text{bytes}}{\text{sample}} \tag{1}$$

$$= \frac{\text{PWs}}{\text{img}} \cdot 128 \cdot 512 \cdot 8 \tag{2}$$

$$= \frac{\text{PWs}}{\text{img}} \cdot 524\ 288\ \text{bytes} \tag{3}$$

where (PWs/img) varies from 1 to 21 PWs per image, (channels/img) = 128, (samples/channel) = 512 and (bytes/sample) = 8 since we have one real and one complex four-byte signed integer for every sample. Arguably, we could have used 2-byte signed integers, but because of limitations in the serialization code using 4-byte integers resulted in faster serialization. See details on the serialization in Section II-C.

We denote the term *the probe* throughout this paper, when referring to the part of our setup mimicking the wireless probe, thus the Verasonics Vantage 256 research scanner recording and processing the raw channel data from the L7-4 probe into a data packet to be transmitted over the network.

B. Mobile Client

In this study, we worked with a range of iOS-based devices, namely, the iPhone 6s, 8, X, and the iPad Pro. The technical specifications of the devices can be seen in Table I. Apple's iOS devices have adopted a unified memory architecture, which allows the GPU and the CPU to access the same physical memory without needing to transfer data between the processors. This is in contrast to the personal computer model of discrete memory where CPU/GPU memory has distinctly separate entities. The unified memory model simplifies the CPU/GPU processing pipeline by allowing the data to be written to a shared CPU/GPU buffer. The GPU compute shader has full access to this buffer and can, thus, process the data without needing to transfer the data to the GPU first. This is in contrast to how some vendors use the term unified memory only meaning a shared address space. A shared address space is convenient to the programmer, but does not improve the performance since copying the data from CPU to GPU occurs behind the scenes.

In addition to the advantage of shared memory architecture, more recent iterations of Apple's hardware platform have inverted the relationship between screen and GPU refresh with what is marketed as ProMotion [38]. Traditionally, the screen receives GPU updates at a fixed frequency, for example, 60 times a second, but with ProMotion, the GPU tells the screen when to update. ProMotion, available on iPad Pro, allows a variable refresh rate from 24 to 120 Hz. A variable refresh rate directly benefits computationally intensive tasks such as beamforming that might have taken too long to complete within 16.67 ms [(1/60) of a second] time budget and high-frame-rate tasks that would be limited by hardware imposed upper bound of 60 Hz.

We will interchangeably use the term *the phone* for all devices, unless the certain device is explicitly stated, throughout this paper.

C. Wireless Networking

For wireless communication, we connected *the phone* and *the probe* over an isolated 802.11ac Wi-Fi network using a Unifi ac-HD-Pro access point (Ubiquiti Networks) to a switch that connects to the scanner and a Dynamic Host Configuration Protocol server via Ethernet. The network pipeline is illustrated in Fig. 2. The base station was configured to provide a wireless connection in the 5-GHz spectrum and with 80-MHz wide channels. The performance was measured in the 400–500-Mbit/s range using iPerf [39]. The Verasonics scanner has an *external* function giving real-time access to the buffer with the raw ultrasound IQ-sampled channel data in MATLAB (The

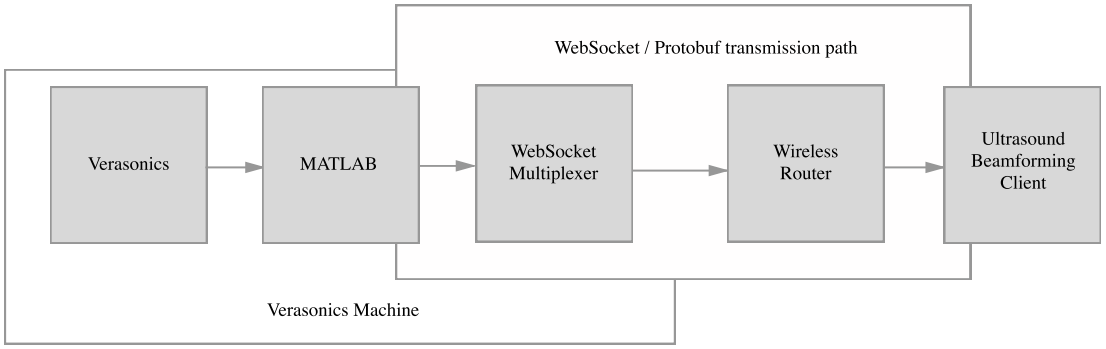


Fig. 2. Flow of data recorded from the L7-4 ultrasound probe connected to the Verasonics Vantage research scanner (*the probe*) to the client (*the phone*).

MathWorks Inc., Natick, MA, USA) once the channel data has been received. The channel data are serialized using Google's protocol Buffers (protobufs) [40] through a Java-based (Oracle Corporation, Redwood Shores, CA, USA) library providing a compact binary message structure to encode the settings and data. It was limitations in this Java library that forced us to use 4-byte integers instead of 2-byte as mentioned in Section II-A. The first protobuf transmitted from *the probe* to *the phone* contains the settings detailing the ultrasound acquisition sequence providing necessary variables for the image reconstruction, while the subsequent protobufs provide the raw channel data sent by the Verasonics research scanner.

To establish the connection between *the phone* and *the probe*, a WebSocket (RFC 6455) multiplexer was implemented using Python (Python Software Foundation). The WebSocket is set up using Zeroconf [41] to advertise the correct port and address for clients to connect to, thereby avoiding hard-coded ports and IP addresses in the client implementation. In short, the networking pipeline, as illustrated in Fig. 2, starts off with the raw channel data received in the Verasonics scanner serialized into a protobuf before it is transmitted via the WebSocket through the wireless router to *the phone*.

D. Beamforming in Software

This section gives a general explanation of the beamforming implemented in software, while the iOS specific implementation is detailed in Section II-E.

The flexibility of software beamforming allows pixel-based beamforming. This means that we can define a set of pixels, our image, and beamform the ultrasound data directly to the pixels. The first step is to calculate the time delay for every signal received on every element to every pixel in the image.

Given the transducer geometry, the element width, and pitch, we get the coordinates for our elements, x_m, and then define the coordinate for every pixel in the image (z, x). Given the speed of sound in human tissue, c, and the angle of the PW transmission, α, we can calculate the two-way time delay [29] as illustrated in Fig. 3 by

$$\tau_{tx}(z, x, \alpha) = (z\cos(\alpha) + x\sin(\alpha))/c \tag{4}$$

$$\tau_{rx}(z, x, x_m) = \sqrt{z^2 + (x - x_m)^2}/c \tag{5}$$

$$\tau(z, x, x_m, \alpha) = \tau_{tx} + \tau_{rx}. \tag{6}$$

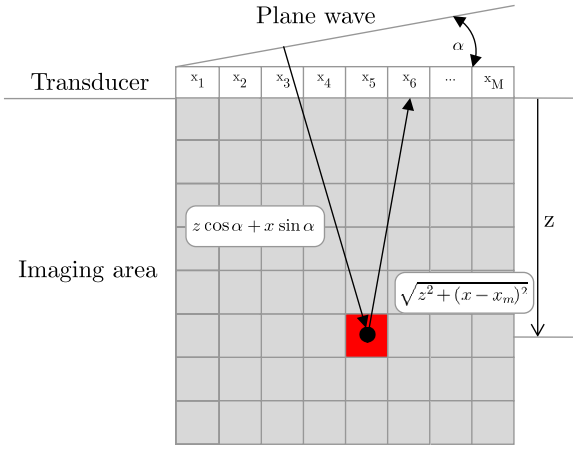


Fig. 3. Illustrating the geometry of the delay calculations in (4)–(6).

The delay calculated in (6) can be a noninteger sample number. To get the correct signal value for every pixel, we linearly interpolate between the samples such that

$$y_{m,a} = \beta u_{m,a}[s] + (1 - \beta)u_{m,a}[s + 1] \tag{7}$$

where $u_{m,a}$ is the signal recorded from element m for transmit a, $s = \lfloor \tau / f_s \rfloor$ is the sample delay calculated from the delay, τ, from (6) and the sampling frequency, f_s, while $\beta = \lceil \tau \rceil - \tau$. For example, if a certain pixel has the delay $\tau = 15.5$ for a sample from a certain element, we simply linearly interpolate between samples 15 and 16 to get the correct value for that pixel using (7).

Each pixel in the image is the sum of the correctly delayed signal

$$I_{\text{DAS}}[z, x] = \sum_{a=0}^{N_\alpha - 1} \sum_{m=0}^{M-1} w_m[z, x] y_{m,a}[z, x] e^{i2\pi f_{\text{demod}} \tau / f_s}. \tag{8}$$

Here, M is the number of elements, $y_{m,a}$ is the correctly delayed IQ-demodulated signal from element m and transmit a, and w_m is given by

$$w_m[z, x] = \begin{cases} 1, & \text{if } |x - x_m| \leq \dfrac{z}{2f\#} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where $[z, x]$ is the pixel position, x_m is the receiving element position, and $f\#$ is the selected ratio between the pixel

depth and the size of the aperture. Thus, w_m results in an expanding dynamic aperture. This is to improve the near-field and have uniform resolution throughout the image. In this study, we used a constant $f\#$ of 1.7.

The multiplication with $e^{i2\pi f_{\text{demod}}\tau/f_s}$ in (8) upmixes the IQ-demodulated signal. f_{demod} is the demodulation frequency, which was equal to the transmitted center frequency, τ is the delay from (6), and f_s is the sampling frequency. The outer sum, where N_α is the number of angular transmits, is the coherent compounding of multiple angular transmits. The correctly delayed complex amplitude for each transmit is accumulated for every pixel.

The final step in our ultrasound image processing chain is to convert the raw complex signal amplitude to a final brightness value in the image. Since the upmixed IQ data are the one-sided analytic signal, the envelope can easily be detected by taking the absolute value of the complex value resulting from the sum in (8). This image magnitude is then converted to decibel values. This is achieved by applying the logarithmic compression

$$I_{\text{dB}} = 20 \cdot \log_{10}(|I_{\text{DAS}}|). \tag{10}$$

The image in decibel is then normalized with relation to a selected maximum value, M_v

$$I_{\text{dB norm.}} = I_{\text{dB}} - M_v \tag{11}$$

before the values outside the desired dynamic range, D_r, are clamped

$$I_{\text{dB final}} = \begin{cases} I_{\text{dB norm.}}, & \text{if } I_{\text{dB norm.}} < -D_r \\ -D_r, & \text{otherwise.} \end{cases} \tag{12}$$

Finally, the image is normalized to gray-level values between [0, 255]

$$I_{\text{grayscale}} = \left\lfloor \frac{255(I_{\text{dB final}} + D_r)}{D_r} \right\rceil \tag{13}$$

where the $\lfloor\ \rceil$ represents a rounding to nearest integer function. The image in $I_{\text{grayscale}}$ can then finally be displayed on the screen.

E. iOS Implementation

The networking and data deserialization on the iPhone is implemented in Swift 4 [42], and the beamforming and image reconstruction is written in Metal [43]. The CPU is responsible for networking and deserialization, queuing up data to be processed by the GPU using a triple-buffer of memory shared by the CPU and GPU. The channel data for each image frame are made available to the Metal GPU compute shader that is run in parallel across the GPU's cores. One compute shader thread is started for every pixel in the image. The number of pixels generated depends on the wavelength of the transmitted pulse chosen, but for an $Z \times X$ number of pixels in an ultrasound image, there are $Z \times X$ threads each summing the contribution (complex sample) of the M number of channels. For the images used in this study, we specifically used a pixel spacing of $\lambda/2$ resulting in $Z \times X = 307 \times 256 = 78\,592$ pixels since the image is 48.8 mm deep, $\lambda = 0.298$ mm, and we used 128 elements with λ spacing. The compute shader implements the beamforming such that it is described in detail in Section II-D. Thus, the associated delay from each receive element to each pixel [see (6)] is calculated *on-the-fly*. Equations (7)–(9) are implemented within a double *for-loop* over the elements M and potentially the multiple transmit angles N_α as described in (8). Equations (10)–(13) are implemented outside the loop with the pixel value from (13) written to the texture, and thus being the only value returned from each compute shader. The code for the iOS app is made available at GitHub (https://github.com/palmerc/Beamformer) with the Metal compute shader implemented in the Beamformer.metal file (Beamformer/SmartWave/Common/Metal/Beamformer.metal).

The client interface allows for real-time dynamic adjustment of the processing variables such as the imaging depth and width, the speed of sound, c, used in (6), the maximum value, M_v, used in (11), and the dynamic range D_r used in (12) and (11). The image is displayed with depth and width rulers, and pinch-and-zoom of the image works as is standard for a touchscreen app.

F. Measuring Frame Rate

We measured the framerate at *the probe* and *the phone* separately by updating a timestamp once for every data packet sent or upon the completion of image formation. If the two framerates were approximately the same, while requiring no more than one frame to be buffered in *the phone's* processing queue, we considered the throughput frames per second (FPS) to be accurate.

The throughput frame rate will vary, and to get a stable estimate, we ran each test for 3 min after the connection between *the probe* and *the phone* was established and the processing of the data had begun.

III. Results

Fig. 1 shows the full setup of the ultrasound probe imaging a CIRS phantom 054GS (CIRS, Norfolk, VA, USA) with the research scanner streaming the channel data wirelessly through the router before it is received, beamformed, and displayed on the iPhone. Fig. 4 shows a screen dump from the iPad Pro of an image created from three angles coherently compounded at 34 FPS. The screen dump also illustrates the width and depth rulers.

Fig. 5(a)–(e) shows, respectively, the ultrasound images dumped from the iPhone 6s imaging cysts with 1, 3, 7, 11, and 21 plane waves with transmission angles uniformly distributed between $\pm 16°$ coherently compounded. Note that the contrast within the hypoechoic cyst is improved with an increased number of angles coherently compounded. Fig. 6 (a)–(e) shows, respectively, the ultrasound images dumped from the iPhone 6s imaging a hyperechoic cyst and point scatterers with 1, 3, 7, 11, and 21 plane waves with transmission angles uniformly distributed between $\pm 16°$ coherently compounded. Note that the resolution of the point scatterers is improved with the increased number of angles coherently compounded.

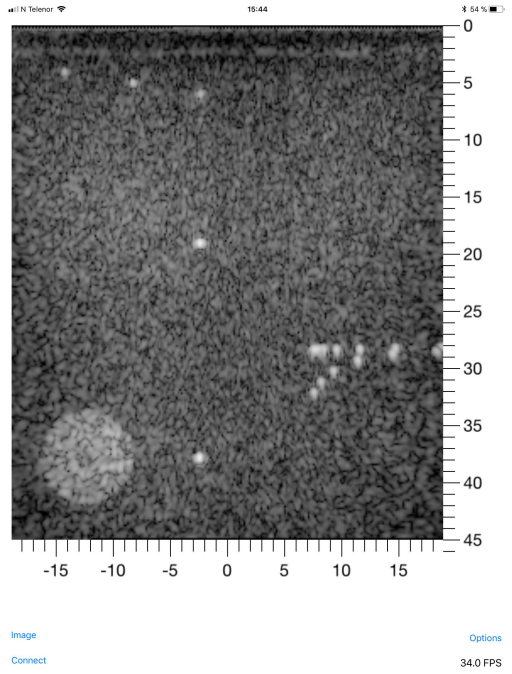


Fig. 4. Screen capture from the iPad processing three angles coherently compounded into a higher quality image at a processing rate of 34 FPS. A demo video is available at https://youtu.be/oN8cwysGxyM.

Fig. 7 shows the timing diagram of the processing in *the probe*, detailing the amount of time spent on, respectively, transmitting and receiving data (Tx+Rx+sampling), serializing the data packet and the amount of time to push the packet into the WebSocket. Note that a significant amount of time is spent serializing the data. Also, note that we perform throttling on the single PW acquisition with the iPhones to slow down the frame rate to below 60 FPS because of the maximum screen refresh rate of 60 Hz. However, the iPad has, as stated earlier, a variable refresh rate up to 120 Hz, and thus, no throttling was necessary. See Section IV for more details.

Fig. 8 shows the timing diagram of the processing in *the phone*, detailing the amount of time spent waiting for a new packet to arrive, receiving the packet and deserializing it, and the time portion spent in the GPU processing the data. Note that GPU processing is not consuming the largest share of the time, and it is actually waiting for receiving and deserializing the web packet that takes most of the time.

Fig. 9 plots the frame rates achieved for different numbers of angles. As expected, higher frame rates are achieved when processing and compounding images with fewer angles. Frame rates of roughly 60 FPS are achieved for all the iPhones tested in the single-angle case, and nearly 90 FPS for the iPad Pro (see Section IV for details on this particular result). A frame rate of approximately 9 FPS is achieved on most devices while compounding up to 11 plane waves. The exception is the iPhone 6s where we saw performance drops off faster than for the other devices.

In Fig. 10(a), the average size of the data packet for each angle transferred over the network and received for processing on the device is plotted both before and after serialization.

Fig. 10(b) shows the amount of data processed per second in the system, thus taking the network packet size shown in Fig. 10(a) and multiplying it by the frame rate achieved by the iPhone X. Note that the data processing rate is equal for all angles except for the single angle case, which was lower because of the throttling necessary to keep the frame rate below 60 FPS.

Also, note that a demo video is available at https://youtu.be/oN8cwysGxyM.

IV. Discussion

The setup described in our study is capable of handling real-time ultrasound imaging with the data wirelessly transmitted from *the probe* to *the phone* doing all processing from IQ-demodulated and downsampled channel data to the full reconstructed image. This is illustrated in Fig. 9, where we can observe that real-time frame rates are achieved for all devices. Fig. 7 illustrates that most of the processing time in *the probe* is spent serializing the data, and only a fraction of the time is spent actually transmitting and receiving the ultrasound signal (Tx+Rx+Sampling). When comparing Figs. 7 and 8, we see that the frame rates achieved are approximately similar and thus, from Section II-F, we can trust the framerates measured. A conventional definition of real-time computing is a system able to finish a task before the scheduled deadline [44]. If we define the deadline in our system as when the channel data for the next image frame arrives on *the phone*, we are doing real-time processing, since *the phone* is processing the data as fast as *the probe* can push data over the network. However, by closely investigating Fig. 8, we can observe that a significant amount of time is spent on waiting, receiving, and deserializing the data, while a smaller portion is spent in the GPU processing the data. This observation indicates that the bottleneck in the system is the serialization of the data in *the probe*.

There is little latency in the system, which means that the image produced on *the phone* is very responsive when moving the probe back and forth on the phantom, as demonstrated in the video available at https://youtu.be/oN8cwysGxyM.

The implementation of the serialization could potentially be improved. Our implementation is done in software calling a Java library from MATLAB since the Verasonics Vantage research scanner only gives access to the channel data buffers through MATLAB. An implementation in a lower level language, or a hardware implementation, accessing the buffers directly could possibly speed up the serialization avoiding the bottleneck in the current setup. Using a different serialization type other than Protobufs could potentially solve the issue with serialization.

The average network packet size, displayed in Fig. 10(a) increases as expected when the data from more angles are included in the packet. When we calculate the average amount of data processed per second in Fig. 10(a), we see that we are transmitting and processing less than 20 MB/s of serialized data (approximately 50 MB/s unserialized). The theoretical bandwidth of the 802.11ac wireless router is 1733 Mbit/s (216.62 MB/s). The bandwidth value is highly dependent on the surrounding RF environment and the type of wireless

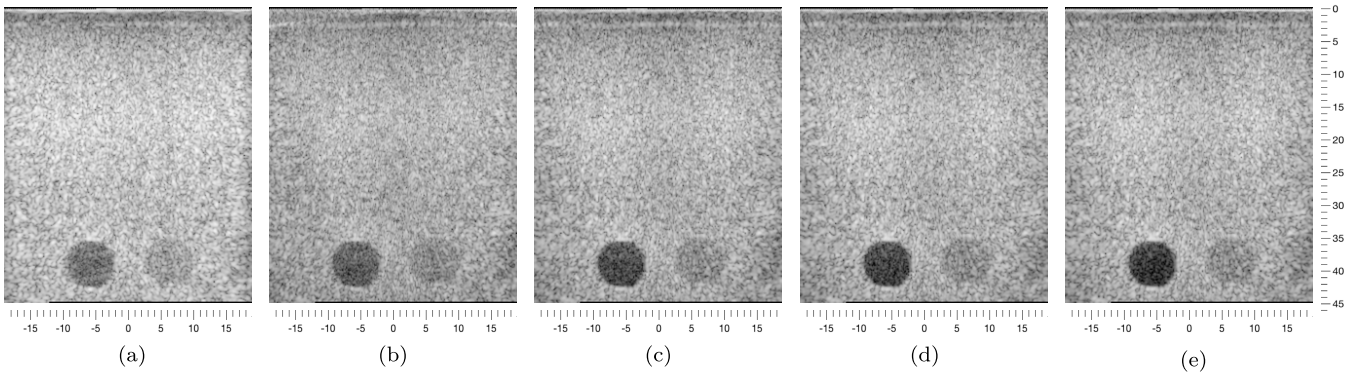


Fig. 5. Ultrasound images dumped from the iPhone 6s imaging cysts with (a) 1, (b) 3, (c) 5, (d) 11, and (e) 21 plane waves with transmission angles uniformly distributed between $\pm 16^{\circ}$ coherently compounded, except the single plane wave transmitted at 0°. Note that the contrast within the hypoechoic cyst is improved with the increased number of angles coherently compounded. The images are displayed with 60-dB dynamic range. A demo video is available at https://youtu.be/oN8cwysGxyM

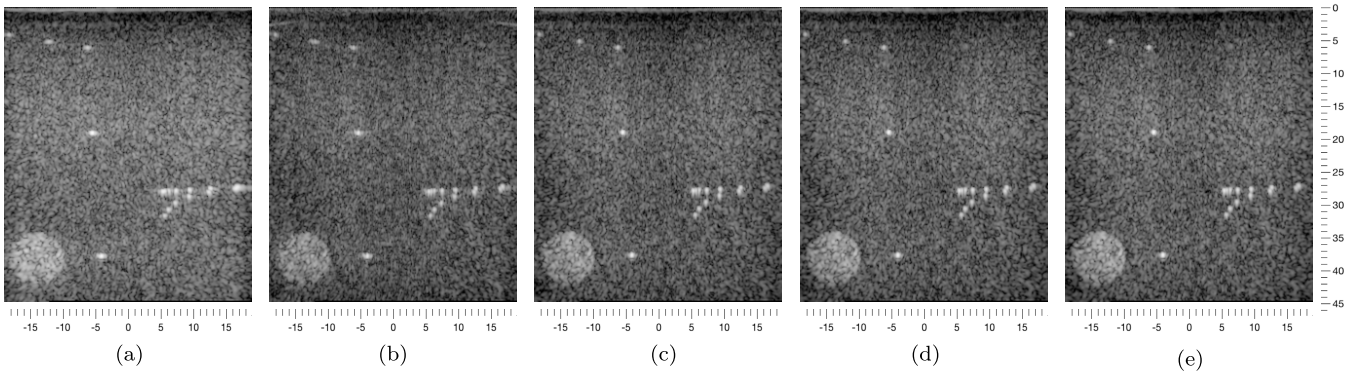


Fig. 6. Ultrasound images dumped from the iPhone 6s imaging a hyperechoic cyst and point scatterers with (a) 1, (b) 3, (c) 7, (d) 11, and (e) 21 plane waves with transmission angles uniformly distributed between $\pm 16^{\circ}$ coherently compounded, except the single plane wave transmitted at 0°. Note that the resolution of the point scatterers is improved with the increased number of angles coherently compounded. The images are displayed with 60-dB dynamic range. A demo video is available at https://youtu.be/oN8cwysGxyM

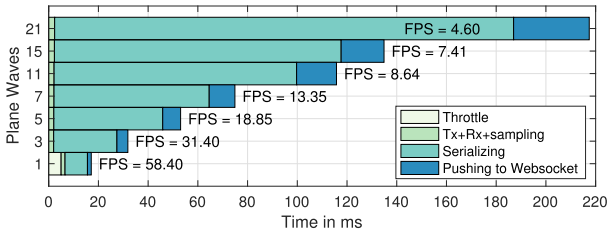


Fig. 7. Timing diagram of the processing in the probe when transmitting to the iPhone, detailing the amount of time spent on, respectively, transmitting and receiving data (Tx+Rx+sampling), serializing the data into the packet, and the amount of time to push the web packet to the WebSocket. Note that when transmitting one PW, we had to throttle the acquisition to get an FPS lower than 60, since the screen refresh rate for the iPhone is maximum 60 FPS. This was not needed when transmitting for the iPad Pro.

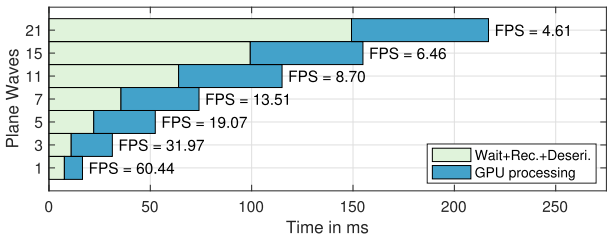


Fig. 8. Timing diagram of the processing in the device, measured for the iPhone X, detailing the amount of time spent on waiting for a new packet to arrive, receiving the packet, and deserializing it, and the portion spent in the GPU processing the data.

hardware on the client. In a real RF environment, we cannot expect to sustain more than 400 Mbit/s in good conditions. Our setup was measured, using iPerf [39], to perform in the 400–500-Mbit/s range. However, we are transmitting less than 20 MB/s (160 Mbit/s) using less than 10% of the theoretical bandwidth, once again indicating that the serialization of the data is the bottleneck in the pipeline and not the wireless networking.

Note that the iPad had a significantly higher processing frame rate for a single PW, while the iPhone X, 8, and 6s was limited to approximately 60 FPS. This is because only the iPad has the capability of screen refresh rates higher than 60 FPS using the ProMotion technology as mentioned in Section II-B. Therefore, we had to throttle *the probe* by adding an additional delay in the Tx+Rx+sampling part of the processing to match the fixed screen refresh rate of 60 FPS when transmitting to the iPhones. If this is not done, the devices are not able to keep up the real-time processing since the data starts to buffer, and a large and growing latency in the processing chain is introduced. This also explains the dip in the first bar in Fig. 10(a), indicating less data processed per second for

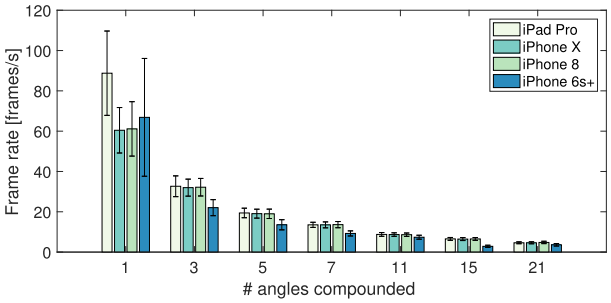


Fig. 9. Throughput processing frame rate as measured on the devices indicated by the legend. These frame rates are made sure to approximately correspond to the frame rate measured in the probe as detailed in Section II-F. As expected, the frame rate drops as the number of angles that are processed and compounded increases.

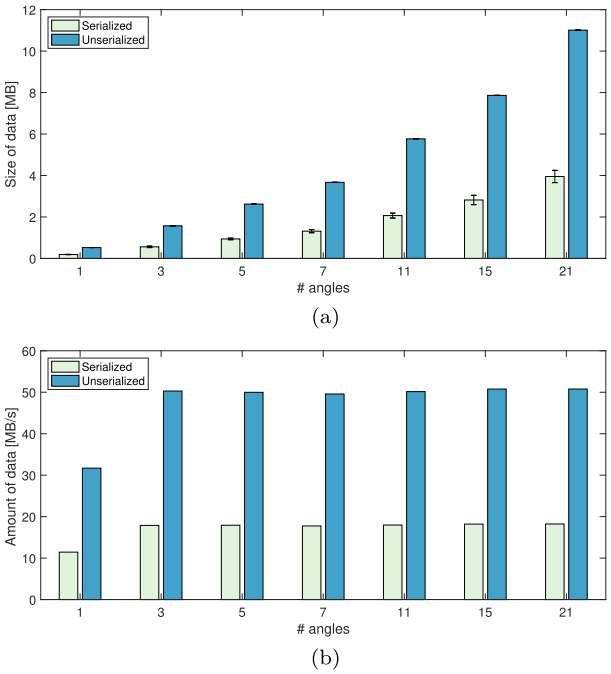


Fig. 10. Size of the data packet for each angle transferred over the network and received for processing on the device is plotted both before and after serialization in (a). The amount of data processed per second in the system, thus when taking the average network packet size from (a) and multiplying it by the frame rate achieved by the iPhone X is shown in (b). The data processing rate is equal for all angles except for the single angle case, which was lower because of the throttling necessary to keep the frame rate below 60 FPS. The large difference between the size of the serialized and unserialized data is most likely because limitations in the Java-based serialization library forced the use of 4-byte integers, where arguably 2-byte integers would be sufficient. Thus, much of the data before serialization was redundant. (a) Average network packet size for the different number of angles. (b) Data processing rates for the different number of angles.

the one angle case. However, for the case of the iPad Pro, the processing was equal for all angles.

It is also worth pointing out that we are not observing any processing improvements between the iPad Pro, the iPhone X, and iPhone 8 when more than one angle is compounded, even though the iPhone X and iPad Pro have a different GPU compared to the iPhone 8. This is probably because the bottleneck in the serialization in *the probe* leads to a significant amount of the processing time in *the phone* is spent waiting and thus the GPU performance is not a differentiating factor for the frame rates achieved. This means that there is a surplus of GPU processing power, a surplus one can expect to increase with the next-generation of iOS devices. If the serialization can be improved this surplus can be spent on compounding more angles at a higher frame rate, or one could potentially exploit the surplus by implementing more sophisticated beamforming techniques, such as adaptive beamforming, for higher quality ultrasound images. By having the software beamforming fully implemented on an *off-the-shelf* device, one can exploit improvements in the hardware, achieving higher quality ultrasound images by simply updating the software in *the phone* while still receiving the same data in *the probe*.

Heating of *the phone* might be an issue when processing extensively on the CPU and GPU. We ran our test for 3 min on each device and did not observe any issues with heating, except for the iPhone 6s where we, as mentioned, saw some performance drop that could be because of heating. However, since, from Fig. 8, most of the processing time in *the phone* is spent waiting, receiving, and deserializing the data in the CPU before the GPU starts to process neither the GPU nor the CPU are run consecutively—reducing the risk of throttling because of heating. The issue with heating is also reduced in the latest generation of devices where the performance is improved dramatically, and also the "sustained performance" without heating is greatly improved [45].

We chose to use a wireless connection, first and foremost, because a wireless connection gives the most portable and durable device. Also, with technologies such as Wi-Fi Direct [46], a high-speed connection between portable devices such as an ultrasound probe and *the phone* is possible. Wi-Fi Direct is known to have transfer rates up to 250 Mbit/s (31.25 MB/s) [47] and, thus, could support our current transfer rate. Whether such transfer rates are possible between two hand-held devices are somewhat uncertain. However, we believe it is safe to speculate that the transfer rates of Wi-Fi Direct will become higher in the future exploiting new Wi-Fi standards such as 802.11ac, as well as miniaturization of improved technology. Wi-Fi Direct is in fact already in use in the hand-held Clarius ultrasound scanner [21], using it to transfer the beamformed ultrasound image to a smartphone to be displayed [48]. A cabled connection could, of course, also provide higher bandwidths with technologies such as USB 3.0 with transfer rates up to 5 Gbit/s (625 MB/s). A wired connection, in addition to the drawbacks of limiting movement of the probe and potentially broken cables, requires specific drivers and potentially licensing for each device the probe can be connected to. Choosing a wireless transfer also fits into current technology trends with more and more devices becoming wireless [49].

The image quality in Figs. 5 and 6 confirms that the image quality is improved, in terms of contrast and resolution, with the number of plane waves coherently compounded. However, the image quality is not improved linearly with the number of angles, the improvement is dominated more with the inclusion of the first angles, thus going from a single PW to three, while

the improvement is less significant going from, e.g., 11 to 21 PWs. This is in agreement with [29]. Our setup using coherent PW imaging allows a tradeoff between the image quality and frame rate. However, one could argue that by coherently compounding 7 angles, the image quality is good enough for simple, clinically valuable imaging. Since coherent compounding of 7 angles results in a frame rate of 13 FPS, one can argue that we are capable of doing real-time imaging with clinically valuable image quality in clinical cases of slow-moving body parts such as the thyroid, the eye, lymph nodes, and some abdominal scanning. However, we also expect to see improvements in both image quality and frame rates with technological improvements in processing power, transfer rates, and more sophisticated beamforming techniques.

V. Conclusion

We have demonstrated real-time, CPWC, ultrasound imaging with the processing handled by *off-the-shelf* devices such as the iPhone and iPad Pro. The frame rates achieved are dependent on the number of angles compounded, resulting in a tradeoff between image quality and frame rate. Frame rates of $\approx$ 60 FPS were achieved for single-PW imaging on the iPhones, while $\approx$ 90 FPS was achieved for the iPad Pro. When compounding 7 angles, a frame rate of 13 frames per second was achieved on all devices tested (iPad Pro, iPhone X, and iPhone 8), except for the iPhone 6s+ which only achieved 9 FPS.

The bottleneck in the current system was identified as the serialization of the data in *the probe*, thus higher frame rates are possible with the current setup by improving the data serialization performance. However, the frame rates and image quality achieved might be good enough for simpler medical clinical ultrasound imaging of, e.g., the carotid artery or thyroid. Since the beamforming can be moved to an *off-the-shelf* device this can reduce the specialized hardware necessary and lower the price of a hand-held system and aiding the democratization of medical ultrasound imaging.

Acknowledgment

The authors would like to thank A. Austeng, Professor at the University of Oslo, Oslo, Norway, for support and technical discussions during this study, and especially for proofreading and improving the paper. They would like to thank T. Hansen, Senior Engineer at the University of Oslo, for assisting in providing and setting up Apple products making this research possible. They would also like to thank Promon AS, Oslo, for providing Apple devices and the wireless equipment for this study.

References

[1] S. Lafitte *et al.*, "Validation of the smallest pocket echoscopic device's diagnostic capabilities in heart investigation," *Ultrasound Med. Biol.*, vol. 37, no. 5, pp. 798–804, May 2011.

[2] C. Prinz and J.-U. Voigt, "Diagnostic accuracy of a hand-held ultrasound scanner in routine patients referred for echocardiography," *J. Amer. Soc. Echocardiogr.*, vol. 24, no. 2, pp. 111–116, Feb. 2011.

[3] E. Godt. (2015). *Handheld Ultrasound a Heartbeat Away From Ousting the Stethoscope?* [Online]. Available: http://www.cardiovascularbusiness.com/topics/imaging/handheld-ultrasound-heartbeat-away-ousting-stethoscope

[4] Vivid E95 cSound. *General Electric Healtcare, Vingmed Ultrasound*. Accessed: Jan. 6, 2018. [Online]. Available: http://www3.gehealthcare.com/en/products/categories/ultrasound/vivid/vivid_e95

[5] Supersonic Imagine. *AiXplorer*. Accessed: Jan. 15, 2019. [Online]. Available: https://www.supersonicimagine.com

[6] O. M. H. Rindal, S. Aakhus, S. Holm, and A. Austeng, "Hypothesis of improved visualization of microstructures in the interventricular septum with ultrasound and adaptive beamforming," *Ultrasound Med. Biol.*, vol. 43, no. 10, pp. 2494–2499, Oct. 2017.

[7] M. A. L. Bell, R. Goswami, J. A. Kisslo, J. J. Dahl, and G. E. Trahey, "Short-lag spatial coherence imaging of cardiac ultrasound data: Initial clinical results," *Ultrasound Med. Biol.*, vol. 39, no. 10, pp. 1861–1874, Oct. 2013.

[8] A. Wiacek *et al.*, "Robust short-lag spatial coherence imaging of breast ultrasound data: Initial clinical results," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 66, no. 3, pp. 527–540, Mar. 2019.

[9] G. Matrone, A. S. Savoia, G. Caliano, and G. Magenes, "The delay multiply and sum beamforming algorithm in ultrasound B-mode medical imaging," *IEEE Trans. Med. Imag.*, vol. 34, no. 4, pp. 940–949, Apr. 2015.

[10] P.-C. Li and M.-L. Li, "Adaptive imaging using the generalized coherence factor," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 50, no. 2, pp. 128–141, Feb. 2003.

[11] J. Camacho, M. Parrilla, and C. Fritsch, "Phase coherence imaging," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 56, no. 5, pp. 958–974, May 2009.

[12] D. Hyun, G. E. Trahey, and J. Dahl, "A GPU-based real-time spatial coherence imaging system," *Proc. SPIE*, vol. 8675, Mar. 2013, Art. no. 86751B.

[13] J. P. Åsen, J. I. Buskenes, C.-I. C. Nilsen, A. Austeng, and S. Holm, "Implementing capon beamforming on a GPU for real-time cardiac ultrasound imaging," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 61, no. 1, pp. 76–85, Jan. 2014.

[14] B. Y. S. Yiu and A. C. H. Yu, "GPU-based minimum variance beamformer for synthetic aperture imaging of the eye," *Ultrasound Med. Biol.*, vol. 41, no. 3, pp. 871–883, Mar. 2015.

[15] T. Su, D. J. Yao, D. Y. Li, and S. Zhang, "Ultrasound parallel delay multiply and sum beamforming algorithm based on GPU," in *Proc. IET 2nd Int. Conf. Biomed. Image Signal Process. (ICBISP)*, May 2017, pp. 1–4.

[16] E. Boni, A. C. H. Yu, S. Freear, J. A. Jensen, and P. Tortoli, "Ultrasound open platforms for next-generation imaging technique development," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 65, no. 7, pp. 1078–1092, Jul. 2018.

[17] J. Joseph. *N.J. EMS Agency Reviews New Whole-Body Ultrasound Device*. Accessed: Jan. 15, 2019. [Online]. Available: https://emsworld.com/article/1221833/nj-ems-agency-reviews-new-whole-body-ultrasound-device

[18] Butterfly Network. *Butterfly IQ*. Accessed: Mar. 1, 2018. [Online]. Available: https://www.butterflynetwork.com

[19] Vscan Family. *General Electric Healtcare, Vingmed Ultrasound*. Accessed: Mar. 1, 2018. [Online]. Available: http://www3.gehealthcare.com/en/products/categories/ultrasound/vscan_family

[20] Siemens Medical Solutions, Ann Arbor, MI, USA. *ACUSON Freestyle Ultrasound System*. Accessed: Mar. 1, 2018. [Online]. Available: https://usa.healthcare.siemens.com/ultrasound/ultrasound-point-of-care/acuson-freestyle-ultrasound-system

[21] *Clarius Wireless Portable Ultrasound*, Clarius, Burnaby, BC, Canada.

[22] Koninklijke Philips N.V. *Lumify*. Accessed: Mar. 1, 2018. [Online]. Available: https://www.lumify.philips.com/

[23] J. Kortbek, J. A. Jensen, and K. L. Gammelmark, "Sequential beamforming for synthetic aperture imaging," *Ultrasonics*, vol. 53, no. 1, pp. 1–16, Jan. 2013.

[24] M. C. Hemmsen *et al.*, "Implementation of synthetic aperture imaging on a hand-held device," in *Proc. IEEE Int. Ultrason. Symp. (IUS)*, Sep. 2014, pp. 2177–2180.

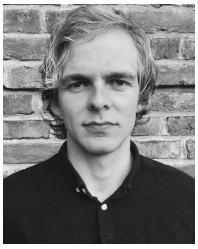
[25] T. D. Ianni, M. C. Hemmsen, P. L. Muntal, I. H. H. Jørgensen, and J. A. Jensen, "System-level design of an integrated receiver front end for a wireless ultrasound probe," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 63, no. 11, pp. 1935–1946, Nov. 2016.

[26] T. D. Ianni *et al.*, "A vector flow imaging method for portable ultrasound using synthetic aperture sequential beamforming," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 64, no. 11, pp. 1655–1665, Nov. 2017.

[27] T. D. Ianni, T. K. Kjeldsen, C. A. V. Hoyos, J. Mosegaard, and J. A. Jensen, "Real-time implementation of synthetic aperture vector flow imaging on a consumer-level tablet," in *Proc. IEEE Int. Ultrason. Symp. (IUS)*, vol. 2, Sep. 2017, pp. 1–4.
[28] M. Tanter and M. Fink, "Ultrafast imaging in biomedical ultrasound," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 61, no. 1, pp. 102–119, Jan. 2014.
[29] G. Montaldo, M. Tanter, J. Bercoff, N. Benech, and M. Fink, "Coherent plane-wave compounding for very high frame rate ultrasonography and transient elastography," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 56, no. 3, pp. 489–506, Mar. 2009.
[30] J.-Y. Lu, "Experimental study of high frame rate imaging with limited diffraction beams," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 45, no. 1, pp. 84–97, Jan. 1998.
[31] D. Garcia, L. Le Tarnec, S. Muth, E. Montagnon, J. Porée, and G. Cloutier, "Stolt's f-k migration for plane wave ultrasound imaging," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 60, no. 9, pp. 1853–1867, 2013. doi: 10.1109/TUFFC.2013.2771.
[32] C. Chen, G. A. G. M. Hendriks, R. J. G. van Sloun, H. H. G. Hansen, and C. L. de Korte, "Improved plane-wave ultrasound beamforming by incorporating angular weighting and coherent compounding in Fourier domain," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 65, no. 5, pp. 749–765, May 2018.
[33] Apple. *Accelerate Framework*. Accessed: Dec. 16, 2018. [Online]. Available: https://developer.apple.com/documentation/accelerate
[34] T. Chernyakova and Y. C. Eldar, "Fourier-domain beamforming: The path to compressed ultrasound imaging," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 61, no. 8, pp. 1252–1267, Aug. 2014.
[35] C. L. Palmer, O. M. H. Rindal, S. Holm, and A. Austeng, "Realtime plane-wave software beamforming with an iPhone," in *Proc. IEEE Int. Ultrason. Symp. (IUS)*, Sep. 2016, pp. 1–4.
[36] H. Hewener and S. Tretbar, "Mobile ultrafast ultrasound imaging system based on smartphone and tablet devices," in *Proc. IEEE Int. Ultrason. Symp. (IUS)*, Oct. 2015, pp. 1–4.
[37] P. Kaczkowski, "Bandwidth sampling data acquisition with the Vantage system for high frequency transducers," Verasonics, Kirkland, WA, USA, White Paper, Jul. 2017, pp. 1–5. [Online]. Available: http://verasonics.com/wp-content/uploads/2017/08/Bandwidth_Sampling_white_paper_July2017.pdf
[38] S. Caldwell. *Apple's Promotion Is Going to Change How We Use Our Devices*. Accessed: Aug. 1, 2018. [Online]. Available: https://www.imore.com/promotion
[39] ESnet and Lawrence Berkeley National Laboratory. *iPerf*. Accessed: Jan. 18, 2019. [Online]. Available: https://iperf.fr
[40] Google, Mountain View, CA, USA. *Protocol Buffers*. Accessed: Aug. 1, 2018. [Online]. Available: https://developers.google.com/protocol-buffers/
[41] IETF Zeroconf Working Group. *Zero Configuration Networking (Zeroconf)*. Accessed: Jan. 18, 2019. [Online]. Available: https://Zeroconf.org
[42] Apple. *Swift*. Accessed: Mar. 1, 2018. [Online]. Available: https://developer.apple.com/swift/
[43] Apple. *Metal*. Accessed: Mar. 1, 2018. [Online]. Available: https://developer.apple.com/metal/
[44] K. G. Shin and P. Ramanathan, "Real-time computing: A new discipline of computer science and engineering," *Proc. IEEE*, vol. 82, no. 1, pp. 6–24, Jan. 1994.
[45] Andrei Frumusanu. *The iPhone XS & XS Max Review: Unveiling the Silicon Secrets*. Accessed: Apr. 10, 2019. [Online]. Available: https://www.anandtech.com/show/13392/the-iphone-xs-xs-max-review-unveiling-the-silicon-secrets/7
[46] Wi-Fi Alliance. *Wi-Fi Direct*. Accessed: Jan. 18, 2019. [Online]. Available: https://www.wi-fi.org/discover-wi-fi/wi-fi-direct
[47] Wi-Fi Alliance. *How Fast is Wi-Fi Direct?* Accessed: Jan. 18, 2019. [Online]. Available: https://www.wi-fi.org/knowledge-center/faq/how-fast-is-wi-fi-direct
[48] K. Dickie, "Wireless ultrasound," Clarius, Burnaby, BC, Canada, White Paper, 2019, pp. 1–4. [Online]. Available: https://clarius.com/education/white-papers/wireless-ultrasound
[49] A. C. Estes. *The Future of Wireless Everything*. [Online]. Available: https://gizmodo.com/the-future-of-wireless-everything-1794814613

Cameron Lowell Palmer (S'07) was born in Dallas, TX, USA, in 1974. He received his B.Sc. and M.Sc. degrees in computer science from the University of North Texas, Denton, TX, USA. He is currently pursuing the Ph.D. degree with the Department of Circulation and Medical Imaging, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.

His research interests include medical applications of augmented reality, real-time mobile beamforming, and automatic quantitative evaluation of diagnostic imagery.



Ole Marius Hoel Rindal (S'14) was born in Norway in 1990. He received the M.S degree in computer science (signal processing) from the University of Oslo, Oslo, Norway, in 2014, where he is currently pursuing the Ph.D. degree with a focus on medical ultrasound beamforming techniques.

He holds a position at the Center for Elite Sports Research, Norwegian University of Science and Technology, Trondheim, Norway, doing research on signal processing on inertial measurement units used in cross-country skiing. His research interests include signal processing, medical image formation, image processing, machine learning, physiological, and biomechanical aspects of cross-country skiing.