Real-time synthetic aperture imaging: opportunities and challenges

Nikolov, Svetoslav; Tomov, Borislav Gueorguiev; Jensen, Jørgen Arendt

Published in:
2006 Fortieth Asilomar Conference on Signals, Systems and Computers

Link to article, DOI:
10.1109/ACSSC.2006.355018

Publication date:
2006

Document Version
Publisher's PDF, also known as Version of record

Citation (APA):
Nikolov, S., Tomov, B. G., & Jensen, J. A. (2006). Real-time synthetic aperture imaging: opportunities and challenges. In *2006 Fortieth Asilomar Conference on Signals, Systems and Computers* (Vol. 1-5, pp. 1548-1552). IEEE. Asilomar Conference on Signals, Systems and Computers. Conference Record https://doi.org/10.1109/ACSSC.2006.355018

Real-time synthetic aperture imaging: opportunities and challenges

Svetoslav I. Nikolov, Borislav G. Tomov, and Jørgen A. Jensen
Center for Fast Ultrasound Imaging, Bldg 348, Ørsted•DTU, Technical University of Denmark,
DK-2800 Lyngby, Denmark

Abstract— **Synthetic aperture (SA) ultrasound imaging has not been introduced in commercial scanners mainly due to the computational cost associated with the hardware implementation of this imaging modality.**

SA imaging redefines the term beamformed line. Since the acquired information comes from all points in the region of interest it is possible to beamform the signals along a desired path, thus, improving the estimation of blood flow. The transmission of coded excitations makes it possible to achieve higher contrast and larger penetration depth compared to ”conventional” scanners.

This paper presents the development and implementation of the signal processing stages employed in SA imaging: compression of received data acquired using codes, and beamforming. The goal was to implement the system using commercially available field programmable gate arrays. The compression filter operates on frequency modulated pulses with duration of up to 50 μs sampled at 70 MHz. The beamformer can process data from 256 channels at a pulse repetition frequency of 5000 Hz and produces 192 lines of 1024 complex samples in real time. The lines are described by their origin, direction, length and distance between two samples in 3D. This parametric description makes it possible to quickly change the image geometry during scanning, thus enabling adaptive imaging and precise flow estimation.

The paper addresses problems such as large bandwidth and computational load and gives the solutions that have been adopted for the implementation.

I. Introduction

This paper describes the possibilities given by synthetic aperture (SA) ultrasound imaging and the problems associated with the implementation of a real-time imaging system.

SA imaging originally developed for radar applications in the 50’s is a radically different method from the common sequential scanning employed in medical ultrasound [1].

Attempts to implement a ultrasound scanner based on SA imaging date back to late 1960s and early 1970s [2], [3]. In the 1970s and 1980s, SA imaging was investigated primarily for nondestructive testing. Although transducer arrays were used, the acquisition method was a direct implementation of radar SA focusing, known as monostatic synthetic aperture imaging [4]. Till the start of the 1990s SA imaging was rarely considered for medical applications. A number of investigations have been conducted in recent years [5]–[11], including some attempts to implement a real-time SA scanner [12]. All these efforts have brought about the creation of novel imaging algorithms such as recursive imaging [13] and the use of orthogonal codes [7], [14]. It has also been demonstrated both with phantoms and *in-vivo* that SA imaging can be used for high-precision blood flow estimation [9], [15], [16]. All these developments are described in greater detail in Section II.

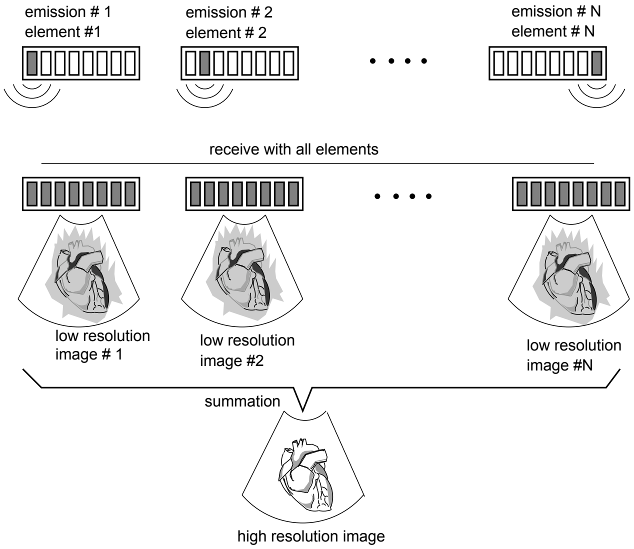

Fig. 1. Basic principle of synthetic aperture ultrasound imaging (from [9]).

Since a number of *in-vivo* investigations have already been successfully carried out, the logical next step is to develop and implement a real-time SA scanner. Rather than developing a scanner that supports one or two imaging modalities, we set about developing a platform, flexible enough to support all of the enumerated algorithms. Section III describes some of the problems that we have encountered in doing so and the choices that we have made.

II. Advantages of SA imaging

SA imaging offers a number of advantages over conventional ultrasound scanning: perfectly focused images, higher-contrast and depth of penetration, fast image acquisition suitable for real-time 3D imaging and precise vector velocity estimates.

A. Better images

The principle of SA imaging is illustrated in Fig. 1. A spherical wave is transmitted using one or several elements. The back-scattered echo stems from the whole region under investigation. All elements are used to record the returned signal and a full image is reconstructed. The image has a low resolution because the transmitted wave is not focused. The measurement is repeated using another origin of the

1-4244-0785-0/06/$20.00

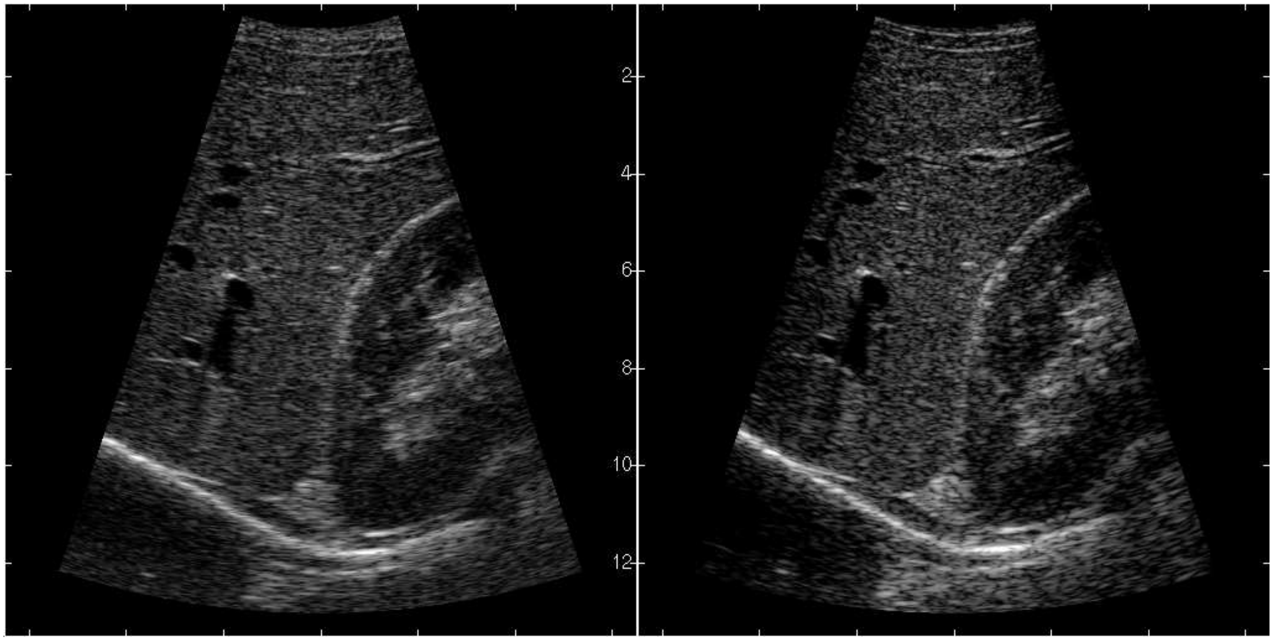

Fig. 2. Example in-vivo images. Left part is the conventional and right is the SA image showing the longitudinal section of right liver lobe showing sections of hepatic vein branches, longitudinal section of a portal vein branch, the kidney and diaphragm at the bottom (from [17]).

spherical wave until the whole aperture is covered. All of the reconstructed images are added together and a high resolution image is created. The beamformation is performed by finding the geometric distance from the origin of the spherical wave to the image point and back to the transmitting element. Dividing this distance by the speed of sound gives the propagation time. The reconstruction process can be expressed as:

$$H(\vec{r}_p) = \sum_{i=1}^{N} a_i(t_p(i)) \sum_{j=1}^{M} a_j(t_p(j)) s_j(t_p(i,j)), \qquad (1)$$

where $\vec{r}_p = [x, y, z]^T$ is the point at which location the image H is being reconstructed, $s_j(t)$ is the signal received by the j:th element, t_p is the propagation time from the transmit location to the point and back to the receive element, and a are the weighting (apodization) coefficients applied on the signal. The propagation time is calculated exactly for every point (see Section III-C) and a perfectly focused image can be reconstructed [9], [17].

A problem in SA ultrasound images is the lower signal-to-noise ratio (SNR) due to the use of un-focused transmit waves. This can be solved by combining multiple element transmissions with frequency modulated signals [9], [18].

A pre-clinical study was conducted by Pedersen, Gammelmark and Jensen [17]. In this study several sonographers were asked to score images with respect to image quality and penetration depth. The SA images had both larger penetration depth and better image quality. Figure 2 shows two screen-shots of the movies presented to the physicians participating in the study.

B. *Better flow estimates*

Color flow maps can be created with high-precision using SA-imaging and cross correlation [9], [15], [16]. Two factors contribute to the precision of flow: the continuous availability of data in all points and the possibility to beamform a signal along the direction of the flow, thus increasing the correlation between signals acquired at different time instances. Figure 3 shows an in-vivo color flow map produced using cross-correlation, directional beamforming and SA imaging. Long FM pulses have been used in transmit resulting in a loss of information in the first 13 mm. A full color flow map can be created in as low as 28 emissions [9], [15] resulting in very high rate. Vector flow imaging has been shown to visualize turbulent flow and combined with the high frame rates of SA imaging, it would be possible to visualize the complex flow patterns in the heart.

C. *Real-time 3D imaging*

The main obstacle for real-time 3D imaging has been the sequential acquisition (line-by-line) of the conventional scanners. Great progress has been made in the area of real-time 3D imaging with the use of 2D matrix transducer arrays. A typical approach is to transmit a broad beam and to form multiple receive beams within its limits [20]. A tradeoff exists, however between frame rate and volume size. The situation becomes worse when blood flow is to be estimated.

SA imaging gives the opportunity to create high-quality volumetric images at high frame rates, while at the same time estimating the blood flow [21], [22] This is enabled by the fact that the number of emissions is not dependent on the image

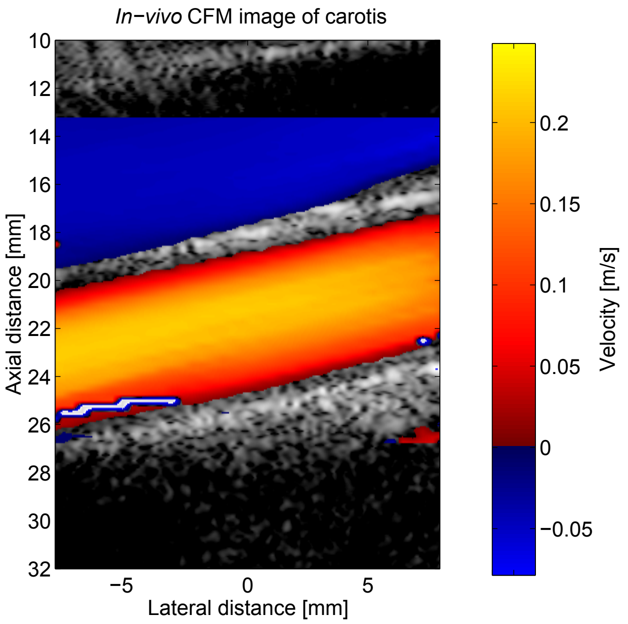

Fig. 3. In-vivo color flow map image at a 77 degree flow angle for the jugular vein and carotid artery. The color scale indicates the velocity along the flow direction, where red hues indicate forward and blue reverse flow (from [19]).

size, and the same acquisitions can be used both for B-mode images and for color flow maps.

III. Real-time implementation

The objective of the project is to build a research scanner that can transmit arbitrary signals, store raw sampled data, and perform conventional and synthetic aperture focusing in real-time. All this should be supported by the same hardware which must be flexible enough to accommodate different imaging geometries.

The basic stages of the scanner are pre-processing, beam-forming, flow estimation and post-processing. The pre-processing stage is responsible for the decoding of the received signals, which in its simplest form consists of a per-channel matched filter. The beam-forming stage consists of a number of focusing units followed by accumulators for the case of SA imaging. The flow estimation is based on cross-correlation. Both I and Q values are produced by the beamformer and the envelope detection, logarithmic compression, scan-conversion and image display are left for a separate PC-based unit [23].

A. Overall system design

The system under development has 1024 channels distributed over 64 boards. The boards are mounted in 19-inch racks in groups of 8. Each board processes data from 16 channels and handles both transmit and receive signal processing. There are 4 Virtex-4 FPGAs on each board for signal processing. A 5th FPGA with a PowerPC processor enabled and Linux operating system running on it, is responsible for parameter setup and communications over a local area network. Inside a board all FPGAs can communicate with each other using serial links. The bandwidth between each 2 FPGAs is 12.8 Gb/sec. The FPGAs can communicate with each other inside each board. The boards are chained so that each of them can communicate with its two neighbors. One of the FPGAs is connected to the analog-to-digital converters and another one is connected to the digital-to-analog converters. These two FPGAs will be denoted as Tx and Rx FPGAs, respectively.

The FPGAs are XC4VFX100 by Xilinx. They have been chosen because of the availability of multi giga-bit serial links, dedicated digital-signal-processing (DSP) slices and the presence of a PowerPC core. There are three distinct types of resources available in these FPGAs: RAM blocks (BRAM), DSP slices capable of single-cycle multiply-and-accumulate operations, and general logic slices. Each device has 376 18-Kbit BRAMs, 160 DSP slices and 42176 logic slices.

The A/D and D/A converters operate at 70 MHz. The processing of samples is performed at double this frequency: $f_{clk} = 140$ MHz. The dedicated DSP and RAM blocks can operate at frequencies of about 400 MHz and will be clocked for any practical purpose at 280 MHz, which leaves a comfortable margin.

B. Matched filtration

The matched filtration is performed in the Rx FPGA, which is responsible for the reception of data, its decimation, storage in external memory, and matched filtration. The data is sent to the other FPGAs via the multi-gigabit serial links.

Present day systems have a penetration depth of about 500 wavelengths (500 λ). SA-imaging can produces image with up to 50% larger penetration when a combination of coded excitations and virtual sources is used. A choice has been made to limit the system to 1000λ. Using a sampling frequency which is 4 times the center frequency, a maximum of 4000 samples are sampled per channel at every emission. Frequency modulated signals are used in transmit. Their length is up to 20 μs, and the transmit clock frequency is 70 MHz. Thus, a waveform is up to 1400 samples long. The pulse compression is performed with the time-reversed transmit waveform weighed with a suitable window. The pulse compression is implemented in frequency domain before beamforming in the Rx FPGA. A 8192-point fast Fourier transform (FFT) is used to avoid circular convolution.

The FFT unit is implemented using an intellectual property core by Xilinx (Xilinx LogiCore FFT v3.2). A pipelined version of the FFT algorithm is used as it gives the best performance/resources ratio. The FFT blocks are clocked at 280 MHz. The number of required FFT blocks depends on how many sets of data must be processed per second. A pulse repetition frequency of 4 kHz provides 70000 clock cycles for data processing. A pipelined FFT block needs 8192 clocks to process data from one channel, thus it can process 8 channels per shot. A total of 4 FFT blocks (2 for the forward and 2 for the inverse transform) are required to process data from 16 channels, which is needed for 3D imaging. They will occupy 34% of the logic slices, 26 % of the RAM blocks and 60 % of the DSP slices.

Six FFT blocks are required to process data in real-time from 16 channels at f_{prf} of 5 kHz. They will use 90% of the available DSP slices.

The samples from the FFT block are processed one at a time, thus 2 complex multiplications are performed per sample. Double (ping-pong) buffering is employed as means of communication between the matched filtration unit and the other units along the data path.

C. Beamforming

Delay-and-sum beamforming is used for both conventional and SA imaging. The images are described as sets of lines. Each line is specified in 3-D space with its origin $\vec{r}_o$, direction $\vec{\zeta}$, length L and distance between two samples Δr. In the case of SA imaging $\vec{r}_o$ can be placed anywhere in front of the transducer and can have arbitrary direction to be able to track flow.

In real-time SA-imaging mode, each board processes data from 4 receive channels and produces 192 lines of 1024 samples each. The beamforming is performed on complex data and both I and Q samples are produced. The board produces a partial contribution of the high-resolution image - it produces a high-resolution image for 4 receive channels ($M = 4$ in (1)). The so-beamformed image is sent to the next board in the chain where it gets summed with high-resolution image for other 4 channels and so forth. One of the challenges is how to partition the beamforming. There are two possibilities: (1) to use one FPGA purely for the accumulation of low-resolution images until a high-resolution one is created and (2) to divide the high-resolution image into regions and each region to be processed in a separate FPGA.

5000 low resolution images of 192 lines of 1024 I and Q 16-bit samples must be created per second. The bandwidth required to transfer the low resolution images is 31.5 Gb/s. The bandwidth available in the system is 12.8 Gb/s, which makes this option unfeasible.

It has therefore been decided that each FPGA will produce a part of the high-resolution image. This relaxes the bandwidth requirements by an order of magnitude, but larger amounts of RAM blocks are required internally to store the intermediate results as it is shown in the following paragraphs.

The beamformer contains the following main blocks: delay calculation unit, apodization generator, interpolation unit and line accumulators. The delay and apodization generators supply the interpolation unit with parameters. The interpolation unit creates samples when a sub-sample delay is needed to produce the result. The outputs of the interpolation units are summed together. The resulting sum is multiplied with the transmit apodization coefficient and is accumulated in the output line buffers. All calculations are made on 16-bit data. The line-buffers are 24-bit to decrease rounding noise. The delay-generation unit and the beamformer are described in greater detail in [24], [25].

1) Delay generation: One of the challenges in describing a SA-image is that the notion of "beam" looses its meaning - one can treat a image as a collection of points. Storing delay and apodization coefficients for every point is not practical. More than 5×10^9 sets of parameters must be read. Each set consists of 16-bit delay value and 8-bit apodization coefficient. Thus 15 GByte/s must be read from an external memory. Assuming a double-data-rate 128-bit interface, only 6.4 GByte/sec can be read at an operating frequency of 200 MHz. Thus the parameters must be calculated dynamically in the FPGA.

The delay-and-sum beamformation requires the calculation of the time-of-flight $t_p(i,j)$ from the origin of transmission $\vec{r}_i$ to the point being beamformed $\vec{r}_p$ and back to a receive element $\vec{r}_j$ (see (1). It is given by:

$$\begin{aligned} t_p(i,j) &= \frac{1}{c}|\vec{r}_p - \vec{r}_i| + \frac{1}{c}|\vec{r}_j - \vec{r}_p| \\ &= t_f + t_b, \end{aligned} \tag{2}$$

where c is the speed of sound. As said, the points are uniformly distributed along a line. The forward and backward propagation times t_f and t_b are calculated independently, and only one of them will be considered in the following. An exact recursive equation exists for the squared propagation times t_i^2 and t_{i-1}^2, where i is the index of the point along the line:

$$t_i^2 - t_{i-1}^2 = A + (2i-1)B, \tag{3}$$

where the constants A and B are calculated as:

$$A = 2(x_{oe}\Delta x + y_{oe}\Delta y + z_{oe}\Delta z)/c^2 \tag{4}$$

$$B = (\Delta x^2 + \Delta y^2 + \Delta z^2)/c^2, \tag{5}$$

where x_{oe}, y_{oe}, z_{oe} are the coordinates of the first point in the line relative to the position of the transmitting (or receiving) element. The increment of the coordinates between two successive points along the line is given by $\Delta\vec{r} = [\Delta x, \Delta y, \Delta z]^T$.

Then the square root of the time is found using a recursive equation. The procedure is described in detail in [24].

2) Apodization: The apodization curves are described using piecewise linear approximation with maximum error of 1% of the full scale. A greedy algorithm is used to find the segments off-line. Each segment is specified by a start value, slope and a segment length in samples. The start value and the slope for each segment are encoded using 14 bits and the number of samples is encoded using 8 bits. Although the internal calculations are performed with 14-bit precision, the result is 8-bit, thus introducing up to 0.4% error in the apodization coefficient.

3) Interpolation: Linear interpolation is used to generate samples for time instances that are not integer multiple of the sampling period. The linear interpolation uses the built-in digital signal processing blocks available in the Xilinx Virtex-4 family of FPGAs. Each such block can multiply two 18-bit numbers and accumulate the result in a 48-bit register. The unit must operate at a clock frequency which is twice higher than the clock frequency of the delay generator, because two samples are needed for each output sample. The output sample is calculated as follows:

$$\begin{aligned} \hat{s}[k] &= \alpha s[n+1] + (1-\alpha)s[n] \\ &= s[n] + \alpha(s[n+1] - s[n]) \end{aligned} \tag{6}$$

The index n is the integer part of the delay. It is represented by 12 bits and can address up to 4096 samples. The sub-sample delay α is formed by the 4 least significant bits of the propagation time.

4) Used resources: Each of the beamformation units can beamform 32 lines during one pulse-repetition period. To achieve the goal of processing close to 200 lines, one must use 6 beamformation blocks.

As mentioned previously, there are 3 critical resources in the FPGAs: dual-port RAM blocks, dedicated DSP computational units and logic blocks. The most critical resource is the number of RAM blocks in the FPGA. A single block RAM (BRAM) can store all of the parameters that describe both apodization and delays for a single element for 32 lines. Parameters for 5 channels must be calculated inside a beamformation unit - 4 receiving and one transmitting. Since the transmitting position changes at every transmission, a new set of parameters must be downloaded from an external memory. Double buffering is used in this case too, and the number of BRAMs for parameter storage is 6 per beamformation unit.

Both I and Q signals are processed in parallel. The raw channel data is 4096 samples long with a 16-bit precision. Accounting for the double-buffering operation, a total of 64 RAM blocks is required to store the raw channel data.

The number of samples in a line is limited to 1024 and the number of lines per beamformation unit is fixed to 32. The precision for the accumulation operation is chosen to be 24-bit, although the data processed further in the system is truncated to 16-bit. A total of 96 BRAMs are required for the formation of high-resolution image inside the FPGA. Since the chosen device has 376 BRAMs, only 2 beamformation units can fit in a single FPGA. In contrast to the usage of memory, these units together utilize only 33% of the available computational resource.

IV. Conclusion

This paper has described the the opportunities that SA-imaging offers as well as some of the problems that have been considered in our real-time implementation. SA imaging yields B-mode images with higher quality. The number of emissions is not coupled with the image size and high frames are achievable using sparse transmissions. The imaging is not confined any more to the traditionally used beams in ultrasound imaging - one reconstructs points inside the region under investigation in a manner that fits one's needs. This has been used to obtain precise blood flow estimates, but the areas of application can be extended to tracking of tracking of specific regions, elasticity imaging etc.

There are a number of challenges that must be overcome in the implementation of synthetic aperture imaging in real time. In our implementation two have been the critical resources - communication bandwidth between devices and the number of available built-in block RAMS.

We have shown that it is possible to develop a system for real-time synthetic aperture imaging using commercially available high-end FPGAs.

References

[1] M. Soumekh. *Synthetic aperture radar. Signal processing with MATLAB algorithms*. John Wiley & Sons, Inc., New York, 1999.

[2] J. J. Flaherty, K. R. Erikson, and V. M. Lund. Synthetic aperture ultrasound imaging systems, United States Patent, US 3,548,642, 1967.

[3] C. B. Burckhardt, P-A. Grandchamp, and H. Hoffmann. An experimental 2 MHz synthetic aperture sonar system intended for medical use. *IEEE Trans. Son. Ultrason.*, 21(1):1–6, January 1974.

[4] J. T. Ylitalo. In-vitro study of computed ultrasound imaging method. In *Proc. IEEE Ultrason. Symp.*, pages 1577–1580, 1994.

[5] M. O'Donnell and L. J. Thomas. Efficient synthetic aperture imaging from a circular aperture with possible application to catheter-based imaging. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 39:366–380, 1992.

[6] M. Karaman, P. C. Li, and M. O'Donnell. Synthetic aperture imaging for small scale systems. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 42:429–442, 1995.

[7] R. Y. Chiao, L. J. Thomas, and S. D. Silverstein. Sparse array imaging with spatially-encoded transmits. In *Proc. IEEE Ultrason. Symp.*, pages 1679–1682, 1997.

[8] G. R. Lockwood, J. R. Talman, and S. S. Brunke. Real-time 3-D ultrasound imaging using sparse synthetic aperture beamforming. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 45:980–988, 1998.

[9] S. I. Nikolov. *Synthetic aperture tissue and flow ultrasound imaging*. PhD thesis, Ørsted●DTU, Technical University of Denmark, 2800, Lyngby, Denmark, 2001.

[10] J. A. Jensen, O. Holm, L. J. Jensen, H. Bendsen, S. I. Nikolov, B. G. Tomov, P. Munk, M. Hansen, K. Salomonsen, J. Hansen, K. Gormsen, H. M. Pedersen, and K. L. Gammelmark. Ultrasound research scanner for real-time synthetic aperture image acquisition. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 52 (5):881–891, May 2005.

[11] J. A. Jensen, S. Nikolov, K. Løkke Gammelmark, and M. Høgholm Pedersen. Synthetic aperture ultrasound imaging. *Ultrasonics*, page Accepted for publication, 2006.

[12] C. R. Hazard and G. R. Lockwood. Theoretical assessment of a synthetic aperture beamformer for real-time 3-D imaging. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 46:972–980, 1999.

[13] S. I. Nikolov, K. Gammelmark, and J. A. Jensen. Recursive ultrasound imaging. In *Proc. IEEE Ultrason. Symp.*, volume 2, pages 1621–1625, 1999.

[14] F. Gran and J. A. Jensen. Directional velocity estimation using a spatio-temporal encoding technique based on frequency division for synthetic transmit aperture ultrasound. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, Accepted for publication, 2005.

[15] S. I. Nikolov and J. A. Jensen. In-vivo synthetic aperture flow imaging in medical ultrasound. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, pages 848–856, 2003.

[16] J. A. Jensen and S. I. Nikolov. Transverse flow imaging using synthetic aperture directional beamforming. In *Proc. IEEE Ultrason. Symp.*, pages 1488–1492, 2002.

[17] M. H. Pedersen, K. L. Gammelmark, and J. A. Jensen. In-vivo evaluation of convex array synthetic aperture imaging. *Ultrasound Med. Biol.*, page Submitted, 2006.

[18] K. L. Gammelmark and J. A. Jensen. Multielement synthetic transmit aperture imaging using temporal encoding. *IEEE Trans. Med. Imag.*, 22(4):552–563, 2003.

[19] J. A. Jensen and S. I. Nikolov. Directional synthetic aperture flow imaging. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, pages 1107–1118, 2004.

[20] D. P. Shattuck, M. D. Weinshenker, S. W. Smith, and O. T. von Ramm. Explososcan: A parallel processing technique for high speed ultrasound imaging with linear phased arrays. *J. Acoust. Soc. Am.*, 75:1273–1282, 1984.

[21] S. I. Nikolov and J. A. Jensen. Investigation of the feasibility for 3D synthetic aperture imaging. In *Proc. IEEE Ultrason. Symp.*, pages 1902–1906, 2003.

[22] J. A. Jensen and S. I. Nikolov. A method for real-time three-dimensional vector velocity imaging. In *Proc. IEEE Ultrason. Symp.*, pages 1582–1585, 2003.

[23] S. I. Nikolov, J. P. G. Gonzaléz, and J. A. Jensen. Real time 3D visualization of ultrasonic data using a standard PC. *Ultrasonics*, 41 (6):421–426, 2003.

[24] S. I. Nikolov, J. A. Jensen, and B. G. Tomov. Recursive delay calculation unit for parametric beamformer. In *Proc. SPIE - Progress in biomedical optics and imaging*, volume 6147-13, pages 1–12, 2006.

[25] S. I. Nikolov, B. G. Tomov, and J. A. Jensen. Parametric beamformer for synthetic aperture ultrasound imaging. In *Proc. IEEE Ultrason. Symp.*, page Accepted for publication, 2006.