

High-Performance Embedded Synthetic Aperture Medical Ultrasound Imaging System

Junying Chen^(✉), Diqin Li, and Huaqing Min

Guangzhou Key Laboratory of Robotics and Intelligent Software,
School of Software Engineering, South China University of Technology,
Guangzhou, Guangdong, China
jychense@scut.edu.cn

Abstract. Medical ultrasound imaging technology has a large scale practical applications in medical diagnostics. In medical ultrasonic synthetic aperture imaging, an array of sensors receive ultrasound echoes and process the echoes to gain low-resolution images which are further processed to obtain high-resolution images. Currently the most common processing algorithm to obtain low-resolution images is delay-and-sum method. This paper mainly focuses on implementing synthetic aperture algorithm in high-performance embedded platform and evaluating its performance. In the estimation of the ultrasonic synthetic aperture algorithm, Field II simulator was used to generate the needed digital ultrasound transducer data. The high-performance embedded computing platform with a graphics processing unit was used to build the synthetic aperture solution and gained an $85 \times$ speedup as compared to its single-core embedded processor implementation. Furthermore, the embedded implementation framework we have built can be easily used to build other high-definition medical ultrasound imaging algorithms.

Keywords: High-performance embedded computing · Medical ultrasound imaging · Synthetic aperture · Graphics processing unit · Parallelization

1 Introduction

Medical ultrasound imaging technology has been widely used and has been rapidly developed. This is because of its advantages of real-time imaging capability, low cost and high safety. In a variety of ultrasound imaging techniques, synthetic aperture imaging technology is one of the high-definition imaging technology. By using this technology, the ultrasound scanner can effectively improve imaging precision to obtain high-resolution ultrasound images. Ultrasonic synthetic aperture technique origins from synthetic aperture radar technology. Synthetic aperture radar [1–3] imaging technique is obtained by processing the reception data to make the relative motion of the small aperture radar and the object become a relatively larger synthesized aperture. In medical ultrasound imaging, synthetic aperture imaging technique [4, 5] processes the received echoes from the sensor array to obtain a low-resolution image, thereby repeatedly superimposing the low-resolution images to form a high-resolution image. Currently the most widely used low-resolution image processing algorithm is delay-and-sum algorithm.

High-performance embedded platforms integrated with graphics processing units [6] possess the characteristics of targeted functionality, high computing performance, relatively lower expense, small footprint, and low energy consumption. Such characteristics make the high-performance embedded computing platforms suitable for mobile and portable functional solutions [7]. Therefore, implementing ultrasonic synthetic aperture imaging algorithm on the high-performance embedded computing platform, can help to effectively reduce the cost of building high-definition medical ultrasound imaging systems, and can help to facilitate future development of portable medical ultrasound imaging systems.

2 Medical Ultrasonic Synthetic Aperture Algorithm

Based on the research of medical ultrasonic synthetic aperture algorithm [8–10], this section will describe the details of the algorithm which is closely related to the subsequent design and implementation in the next section.

2.1 Low-Resolution Image Acquisition

The most commonly used algorithm for low-resolution image acquisition is delay-and-sum algorithm. Before the delay apodization is applied, the ultrasound receive channel data is first manipulated with Hilbert transform. The Hilbert-transformed ultrasound data sample is calculated by the following convolution output:

$$\text{conv}_{a,b}(i) = \sum_{j=1}^{i+T-1} h(j) d_{a,b}(j) \quad (1)$$

In Eq. (1), a represents for the a th ultrasound receive channel, b represents for the b th transmit firing, and i represents for the i th ultrasound imaging depth sample, thus $d_{a,b}(j)$ is the j th data sample of a th ultrasound receive channel for b th transmit firing. Furthermore, $h(j)$ denotes the Hilbert transform coefficient which is expressed as:

$$h(j) = \begin{cases} 0 & \text{for even } j \\ \frac{2}{\pi j} & \text{for odd } j \end{cases} \quad (2)$$

Hilbert transform function is similar to a finite-impulse response (FIR) filter with T number of taps, as shown in Eq. (1), where T is usually set as 51. The Hilbert transform process is shown in Fig. 1.

Delay-and-sum beamforming method is applied on the Hilbert transformed data to obtain a low-resolution image for the b th transmit firing. Use $L_b(p)$ to represent the value of pixel p in the low-resolution image of b th transmit firing, according to delay-and-sum algorithm, $L_b(p)$ can be expressed as:

$$L_b(p) = \sum_{a=1}^N w_a \cdot x_{a,b}(p) \quad (3)$$

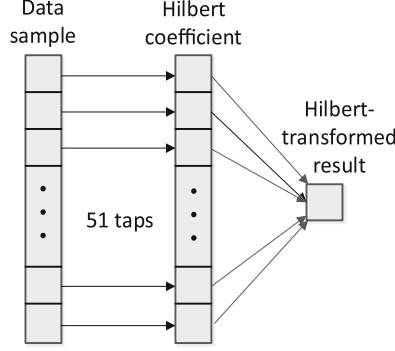


Fig. 1. Hilbert transform process

In Eq. (3), w_a is the apodization weight for the a th ultrasound receive channel which adopts Hanning window function, and $a_{max} = N$. $x_{a,b}(p)$ is the interpolated channel data sample which can be obtained by:

$$x_{a,b}(p) = \gamma \cdot conv_{a,b}(\beta) + (1 - \gamma) \cdot conv_{a,b}(\beta + 1) \quad (4)$$

$conv_{a,b}(\beta)$ and $conv_{a,b}(\beta + 1)$ are the two adjacent Hilbert-transformed data samples corresponding the most closely with the pixel's focusing delay in the a th channel. β is the imaging depth sample index corresponding to the focusing delay $\delta_{a,b}(p)$, as shown in Fig. 2(a), and γ is the interpolation weight (between 0 and 1). For a given sampling rate f_s , β and $\delta_{a,b}(p)$ can be found as:

$$\beta = \text{floor}(f_s \cdot \delta_{a,b}(p)) \quad (5)$$

$$\gamma = 1 + \beta - f_s \cdot \delta_{a,b}(p) \quad (6)$$

In Eq. (5), floor operation outputs the largest integer which is not greater than $(f_s \cdot \delta_{a,b}(p))$. $\delta_{a,b}(p)$ is calculated by:

$$\delta_{a,b}(p) = (D_t(p) + D_r(p)) \div c \quad (7)$$

In Eq. (7), $D_t(p)$ and $D_r(p)$ denote the transmit propagation distance and the receive propagation distance respectively, as shown in Fig. 2(b). c is the ultrasound propagation speed.

2.2 High-Resolution Image Recursive Synthesis

High-resolution image is obtained by recursively synthesize a series of low-resolution images. For k th high-resolution image with a frame size of M , its value of pixel p is calculated as:

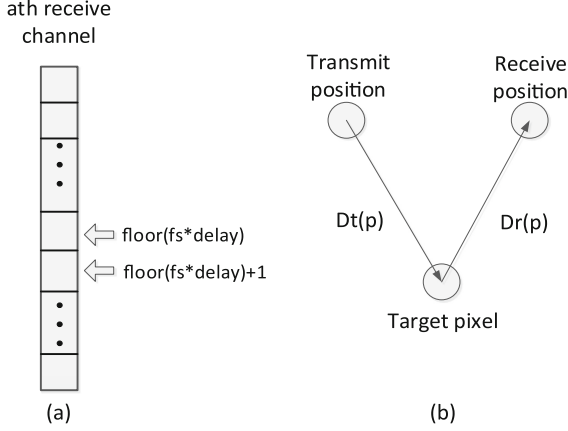


Fig. 2. (a) $\beta = \text{floor}(f_s \cdot \delta_{a,b}(p))$ for the a th ultrasound receive channel, where $\delta_{a,b}(p)$ is the focusing delay; (b) transmit propagation distance and the receive propagation distance.

$$H_k(p) = \sum_{i=k-M+1}^k L_i(p) \quad (8)$$

$L_i(p)$ is the corresponding pixel value from the i th low-resolution image. Equation (8) can be rewritten as:

$$H_k(p) = H_{k-1}(p) + L_k(p) - L_{k-M}(p) \quad (9)$$

In Eq. (9), $H_{k-1}(p)$, $L_k(p)$ and $L_{k-M}(p)$ are the corresponding pixel value for previous high-resolution image, the latest low-resolution image and the earliest low-resolution image.

3 High-Performance Embedded Platform and Algorithm Implementation

3.1 High-Performance Embedded Computing Platform with GPU

Figure 3 demonstrates the hardware architecture of the high-performance embedded computing platform with graphics processing unit. The simulated ultrasound channel data was first transmitted to the random-access memory (RAM) on the embedded computing platform. The synthetic aperture algorithm was performed in the embedded GPU. After the imaging process data was ready for final display, OpenCV library functions were used to display the images on the connected monitor. Finally, the imaging processed data was stored in the storage unit.

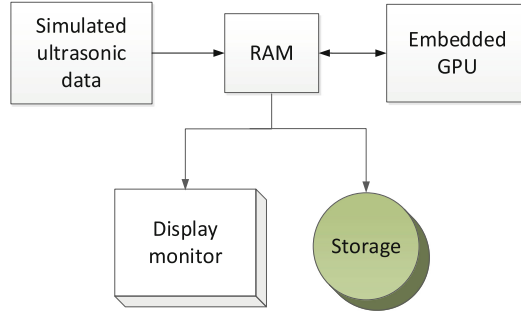


Fig. 3. Architecture of high-performance embedded computing platform

3.2 Implementation of Medical Ultrasound Synthetic Aperture Imaging

This section will describe the implementation methodology of medical ultrasound synthetic aperture imaging on high-performance embedded computing platform. The implementation procedures were conducted based on the algorithm details elaborated in Sect. 2.

Hilbert Transform Implementation. Athread block in the graphics processing unit was assigned to compute one ultrasound receive channel pre-process data. Within the thread block, each computing thread was used to execute one Hilbert transform operation to derive the Hilbert-transformed data samples for an array of consecutive imaging data samples in the receive channel. The processing data sample array was updated time by time to refresh the computing input data sources. This method was used to keep the run-time data storage at an appropriate size, so as to realize a high-efficient parallelization. The Hilbert transform implementation parallelization is shown in Fig. 4.

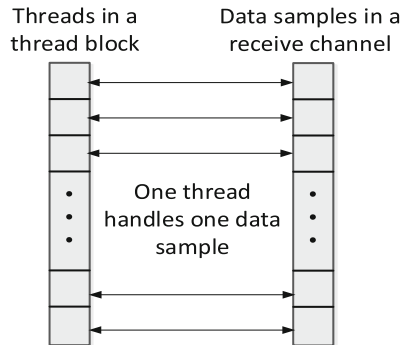


Fig. 4. GPU parallelization for Hilbert transform implementation. A thread block processes one column of 256 data samples at a time, and then repeats.

Delay-and-Sum Algorithm Implementation. Delay-and-sum processing block processes the Hilbert-transformed data samples to generate a series of low-resolution images. To complete this imaging process, the graphics processing unit assigns every thread block to compute the pixel values of the low-resolution images on a two-dimensional GPU compute grid basis, as shown in Fig. 5. Within the thread block, each thread is designed independently to calculate a single low-resolution image pixel value. The computation procedures are deduced from Eqs. (3), (4), (5), (6) and (7). Basically, the computation process is implemented by first calculating the focusing delay of the targeted pixel referring to Eq. (7). Based on the focusing delay value, calculate the imaging depth sample index corresponding to the focusing delay and the interpolation weight, according to Eqs. (5) and (6). Then, compute for the interpolated channel data sample using Eq. (4). Finally, obtain the targeted pixel value of the low-resolution image by applying apodization weights to the interpolated channel data sample as illustrated in Eq. (3).

High-Resolution Image Recursive Calculation. When low-resolution images are produced, they are transmitted to high-resolution processing block to compute for the high-resolution image. As the high-resolution image calculation process is recursive but the calculation of each high-resolution image pixel value is independent with each other, each computing thread in the graphics processing unit can be used to perform one high-resolution image pixel value, and multiple threads conduct the calculation process simultaneously.

Memory Access and Management. Memory utilization is an important part of building such a high-performance embedded medical ultrasound imaging platform. The memory access speed in the high-performance embedded computing platform is register files > local memory > global memory, but the memory capacity distribution is register files < local memory < global memory. As a result, the register files are so precious that only current processing data and results are stored in them. Local memory is usually used for intermediate calculation results, and global memory is used to store initial data and final results.

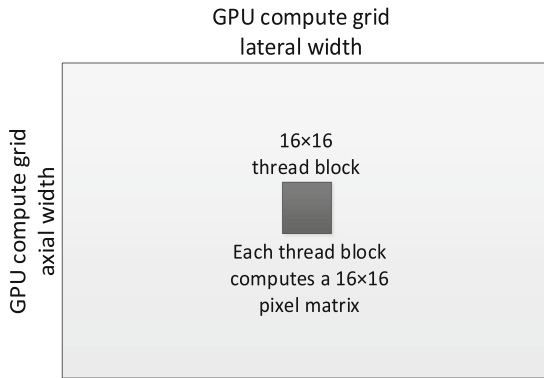


Fig. 5. GPU computing scheme for delay-and-sum algorithm implementation.

4 Experimental Design and Results

4.1 Experimental Environment and Lab Scenario Design

The experiments adopted Field II simulator [11, 12] to simulate the ultrasound channel data samples. The simulation parameters are shown in Table 1.

Table 1. Field II simulation parameters.

Parameter	Value
Ultrasonic speed propagated in human tissue	1540 m/s
Number of transducer elements	128
Element pitch	0.3048 mm
Pulse repetition rate	5 kHz
Sampling rate	40 MHz

The experimental high-performance embedded computing platform is Nvidia Jetson TK1 embedded evaluation platform. The computing resources of this platform are a 4-plus-1 Cortex-A15 ARM processor and an Nvidia Kepler GPU with 192 CUDA computing cores. Such computing core can afford required computing power for the experiments. The architecture details of Jetson embedded computing platform is shown in Table 2.

Table 2. Nvidia Jetson TK1 embedded platform computing parameters.

Parameter	Value
ARM processor clock rate	2.3 GHz
DDR3 memory size	2 GB
GPU processor clock rate	0.85 GHz
Global memory size	1746 MB
L2 cache size	128 KB
Constant memory	64 KB
Shared memory size per block	48 KB
Register files size per block	32768

In the Hilbert transform stage, one thread block take charge of a whole receive channel's data sample processing, and each thread inside the thread block handles the Hilbert transform for each data sample. The size of the thread block is 256, thus if the total size of the receive channel data samples is larger than 256, the computing process repeats. In delay-and-sum processing stage, every thread block calculates the low-resolution image pixel values for a two-dimensional block of 16×16 pixels. Besides, the compute grid of the thread blocks is also two-dimensional, with the lateral

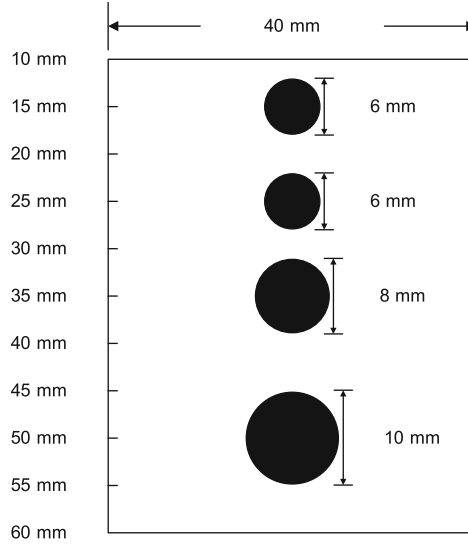


Fig. 6. Performance evaluation experimental scenario

size as one-sixteenth of the image lateral pixel size, and the axial size as one-sixteenth of the image axial pixel size.

Figure 6 illustrates the experimental scenario to evaluate the performance of the imaging algorithm implementation.

4.2 Experimental Results and Discussions

Figure 7 shows the experimental imaging output of the simulated scenario. As shown in Fig. 7, the high-definition quality of the medical ultrasonic synthetic aperture imaging was maintained by this high-performance embedded computing solution.

The computing performance improvement is shown in Table 3. As seen from Table 3, using the 4-core multi-core ARM processor can accelerate the medical ultrasonic synthetic aperture imaging algorithm by nearly $4 \times$ speedup, and using the 192-core GPU can accelerate the imaging algorithm by around $85 \times$ speedup, both compared to the single-core ARM processor's execution timing result.

As a result, accelerating the compute-intensive medical ultrasound synthetic aperture without distortion using high-performance parallel computing technique is a feasible. Thus, such platform can be used to facilitate other high-definition medical ultrasound imaging algorithms' acceleration and the future development of portable medical ultrasound imaging systems.

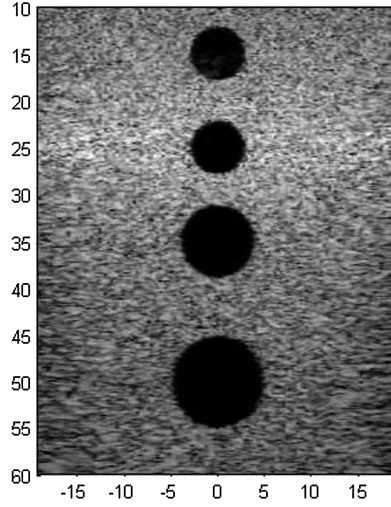


Fig. 7. Experimental output image of simulated scenario

Table 3. Imaging performance for various computing resources (number of operations = 28923002880 Ops).

Computing resource	Execution time (ms)	Throughput (Gflop/s)
Single-core ARM processor	192160.000	0.15
Multi-core ARM processor	51649.500	0.56
GPU processing cores	2282.035	12.68

5 Conclusions

We investigated the acceleration feasibility of compute-intensive medical ultrasound synthetic aperture imaging algorithm on high-performance embedded computing platform with graphics processing unit in this paper. As the data independency of the synthetic aperture imaging algorithm and the high parallelization of the GPU processing units on the embedded computing platform, the embedded GPU cores' computational acceleration is $85 \times$ as compared to the single-core ARM processor. Furthermore, the embedded implementation framework we have built can be easily used to build other high-definition medical ultrasound imaging algorithms, which will be done in our subsequent research.

Acknowledgements. This work is supported by “Guangzhou Science and Technology Program” (Key Laboratory Project, No. 15180007) and “the Fundamental Research Funds for the Central Universities (No. 2015ZM081)”.

References

1. Ossowska, A., Junghyo, K., Wiesbeck, W.: A simulation for synthetic aperture radar with digital beam-forming in elevation. In: IEEE International Conference on Geoscience and Remote Sensing Symposium (IGARSS), pp. 1407–1410 (2006)
2. Krieger, G., Gebert, N., Younis, M., Moreira, A.: Advanced synthetic aperture radar based on digital beamforming and waveform diversity. In: IEEE Radar Conference (RADAR), pp. 1–6 (2008)
3. Rincon, R.F., Fatoyinbo, T., Ranson, J., Sun, G., Perrine, M., Bonds, Q., Valett, S., Seufert, S.: Digital beamforming synthetic aperture radar (DBSAR) polarimetric operation during the Eco3D flight campaign. In: IEEE International Conference on Geoscience and Remote Sensing Symposium (IGARSS), pp. 1549–1552 (2012)
4. Ylitalo, J.T., Ermert, H.: Ultrasound synthetic aperture imaging: monostatic approach. IEEE Trans. Ultrason. Ferroelectr. Freq. Contr. **4**(3), 333–339 (1994)
5. Nikolov, S.I., Jensen, J.A., Tomov, B.G.: Fast parametric beamformer for synthetic aperture imaging. IEEE Trans. Ultrason. Ferroelectr. Freq. Contr. **55**(8), 1755–1767 (2008)
6. NVIDIA Tegra K1: A new era in mobile computing. Nvidia, Corp., White Paper (2014)
7. Wolf, M.: High-Performance Embedded Computing: Applications in Cyber-physical Systems and Mobile Computing. Newnes, Amsterdam (2014)
8. Jong-Ho, P., Changan, Y., Jin-ho, C., Yangmo, Y., Tai-Kyung, S.: A real-time synthetic aperture beamformer for medical ultrasound imaging. In: IEEE Ultrasonics Symposium (IUS), pp. 1992–1995 (2010)
9. Yiu Billy, Y.S., Tsang Ivan, K.H., Yu Alfred, C.H.: GPU-based beamformer: fast realization of plane wave compounding and synthetic aperture imaging. IEEE Trans. Ultrason. Ferroelectr. Freq. Contr. **58**(8), 1698–1705 (2011)
10. Choye, K., Changan, Y., Jong-Ho, P., Yuhwa, L., Hwa, K.W., Min, C.J., Ihn, C.B., Tai-Kyong, S., Yang-Mo, Y.: Evaluation of ultrasound synthetic aperture imaging using bidirectional pixel-based focusing: preliminary phantom and in-vivo breast study. IEEE Trans. Biomed. Eng. **60**(10), 2716–2724 (2013)
11. Jensen, J.A.: Field: a program for simulating ultrasound systems. In: Paper presented at the 10th Nordic-Baltic Conference on Biomedical Imaging Published in Medical & Biological Engineering & Computing, vol. 34, pp. 351–353, Supplement 1, Part 1 (1996)
12. Jensen, J.A., Svendsen, N.B.: Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers. IEEE Trans. Ultrason. Ferroelectr. Freq. Contr. **39**, 262–267 (1992)

Embedded System Technology

13th National Conference, ESTC 2015, Beijing, China,

October 10-11, 2015, Revised Selected Papers

Zhang, X.; Wu, Z.; Sha, X. (Eds.)

2015, X, 189 p. 94 illus. in color., Softcover

ISBN: 978-981-10-0420-9