3D Ultrasonic Signal Compression Algorithms for High Signal Fidelity

Pramod Govindan, Thomas Gonnot, Spenser Gilliland and Jafar Saniie
Department of Electrical and Computer Engineering
Illinois Institute of Technology,
Chicago, Illinois 60616

***Abstract* – Ultrasonic signal processing applications require huge amounts of data to be processed. Further, high computational performance is essential to meet the real-time requirements. Compression of the signal helps to reduce the data size and storage requirements as well as allow for rapid transmission of data to remote locations. High signal fidelity is significant in many of practical applications like ultrasound medical imaging and nondestructive testing. In this study, we discuss two methods for ultrasonic signal compression which offer high signal fidelity – Discrete Wavelet Transform and signal decimation with the Nyquist rate limit. The compression algorithm is implemented on a reconfigurable system-on-chip platform using programmable hardware logic as well as in software using an embedded processor. The implementation details and the performance of the compression algorithms on both the hardware and software are analyzed in this paper.**

I. INTRODUCTION

Ultrasonic systems are widely used in industrial and medical imaging applications for diagnosis, object recognition and classification. One of the major challenges in ultrasonic signal processing applications is the large volumes of data to be processed. Another challenge is the requirement of real-time processing of compute–intensive signal processing algorithms. The acquired data is send to remote locations for analysis via low bandwidth communication channels; so, the data transmission time has to be reduced by compressing the signal as much as possible without degrading the reconstruction quality.

Noise will be generated as part of the sampling process. Some uncertainty in the timing of the sampling interval, particularly the initiation of the sampling process, results in jitter. In addition, the jittering noise will be pronounced when minor geometrical or mechanical variation occurs in the experiment. Sampling at a rate higher than Nyquist will help to remove some of the noise, allowing a good quality signal to be reproduced. This oversampling also helps reduce aliasing. The SNR can be improved in an oversampled signal by averaging out the noise. The oversampling technique can distribute the quantization noise outside the signal band. Since most of the time, the signal is oversampled to get the above benefits; the sampled signal will have a lot of redundant information. This will create storage problem and also will cause a large delay in transmitting the data to a remote location; thus, the signal needs to be compressed. Signal compression is meant to remove the redundancies in the signal. Also the compression involves removing the irrelevant components of the signal. Where the function has more desirable features, we can use higher sampling rate, and where the function is smooth, we can use fewer samples and still get a good quality of signal information.

II. COMPRESSION METHODS

In this section, we present and evaluate two methods for compression for ultrasonic imaging applications, with the objective to maintain higher compression ratio, exceeding a predefined peak signal-to-noise ratio. One method is compression using discrete wavelet transform and the other one is compression using decimation.

A. Compression using Discrete Wavelet Transform

One of the most common and efficient methods for compression is the Discrete Wavelet Transform (DWT). In wavelet based compression, a technique called "sub-band coding" is used. In sub-band coding, the input samples are filtered to reduce the bandwidth of the signal, thus generating a sub-band of the input signal. This will be again subsampled to get a narrower band. The subsampled signal generates smaller size transform coefficients which can be synthesized to reconstruct the original signal. This approach helps to identify the redundant components which are not needed to reconstruct the original signal. Thus the signal becomes frequency localized. A majority of the significant portions of the signal will be localized to a particular frequency region, especially the low frequency region for most of the practical applications of ultrasonic testing.

DWT is a signal filtering process and is equivalent to convolution of the signal with impulse response of the filter as shown:

$$x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k).h(n-k) \quad (1)$$

where h(n) are the lowpass filter coefficients. In the DWT, the resolution is affected by filtering. The scale is changed by subsampling. The lowpass filtering reduces resolution by 2, but scale is unchanged. The subsampling by 2 doubles the scale. The filtering and subsampling is:

$$y(n) = \sum_{k=-\infty}^{\infty} h(k).x(2n-k) \quad (2)$$

In (2), 2n shows subsampling by 2.

978-1-4799-0066-4/13/$31.00 ©2013 IEEE

One level decomposition into lowpass and highpass components is expressed using (3) and (4).

$$y_{high}(k) = \sum_n x(n).\, g(2k - n) \quad (3)$$
$$y_{low}(k) = \sum_n x(n).\, h(2k - n) \quad (4)$$

In each level of decomposition, 2^{j-1} coefficients are generated, j is the level, where 2^{j} is the 'scale', j < J, where 2^{J} is the number of samples in the original signal.

The dilation (or refinement) equation and wavelet equation are given by (5) and (6) respectively.

$$\Phi(t) = \sum \sqrt{2}\; h(n)\; \Phi(2t - n) \quad (5)$$
$$\psi(t) = \sum \sqrt{2}\; g(n)\; \Phi(2t - n) \quad (6)$$

where h(n) are the lowpass filter coefficients and g(n) are the highpass filter coefficients. The lowpass filter determines the scaling function Φ(t). The highpass filter determines the wavelet function ψ(t). The wavelet filter is the mirror reflection of the scaling filter with alternating signs. For example, if the scaling filter coefficients are h_k = (h1, h2, h3, h4), then the wavelet filter coefficients are g_k = (h4, -h3, h2, -h1). Different wavelet basis can be generated by using different filter coefficients h & g [4] [5].

A volumetric image of 128*128*2048 ultrasonic data samples is used for the compression analysis. Fig. 1 represents such a 3D block of data. This experimental data is generated using a 2 inch by 2 inch steel block sample. Data are sampled at a rate of 100 MHz by using a 5 MHz transducer. This oversampling is done to retain very fine features of the ultrasonic signal. Each measurement (i.e., A-scan) has 2048 samples (An example of A-scan can be seen in Fig. 6a). A total of 128 such A-scans are taken per line in y-direction. These 128 A-scans are repeated 128 times in z-direction to get the 3D block of data (Fig. 1).

3D ultrasonic data compression is implemented by applying 1D-DWT for x, y and z coordinates. A-scans will have a significant amount of redundant information due to the oversampling. To get maximum compaction of the A-scans, a high-order wavelet Daubechies-10 (db10) is used to compress the signal in the x-direction. Our experiments show that Daubechies wavelet basis is more compatible for the ultrasonic signal compression because of the similarity between the Daubechies wavelet function (Fig. 2) and the ultrasonic signal echo [9].

In the x-direction, the packet decomposition method is used as shown in Fig. 3. Here, apart from further decomposing the lowpass (L) components, selected highpass (H) components with higher energy distribution

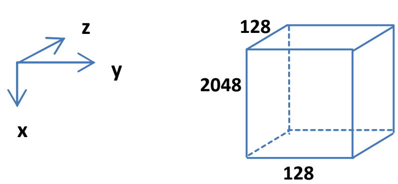

Fig. 1. 3D block of ultrasonic data.

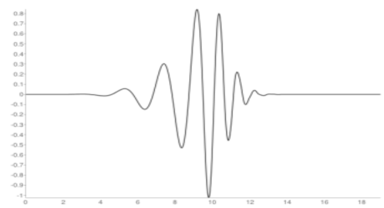

Fig. 2. Daubechies -10 wavelet function

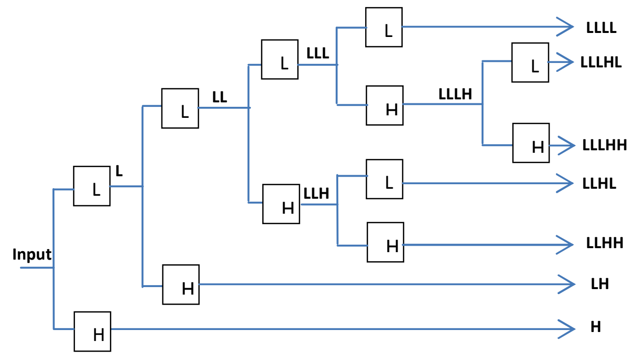

Fig. 3. Wavelet packet decomposition in x-direction

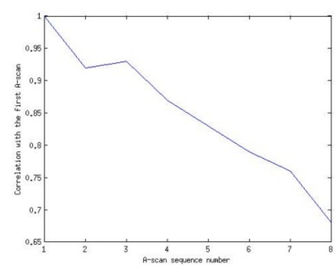

Fig. 4. Correlation between neighboring A-scans

are also decomposed to improve the data compaction, which leads to a higher compression ratio [2]. But the reconstruction is complex since the non-zero wavelet coefficients are spread between lowpass and highpass regions.

For the y and z directions, we utilize the correlation properties of the neighboring A-scans [7]. Fig. 4 shows the correlation plot of 8 neighboring A-scans in a line in y-direction (This is similar in z-direction also). The correlation between the first A-scan and all the 8 A-scans are calculated. It can be observed from Fig. 4 that there is a high correlation between the neighboring A-scans. So it is clear that there are some redundancies between nearby A-scans, which can be removed. To reduce the computation time, a simple Haar kernel is used for DWT in y and z directions.

According to Fig. 3, the DWT coefficients {LLLL, LLLHL, LLLHH, LLHL, LLHH, LH, H} for one particular A-scan is plotted in Fig. 5. By discarding the low energy coefficients (H, LH & LLHL), 79% compression is achieved in x-direction. By applying Haar wavelet in y & z direction, an additional 75% reduction of the compressed results from x-direction is obtained because of the low energy content of H & LH coefficients in y & z direction. Thus, the total 3D ultrasonic data compression is 98.7%. Fig. 6 shows the original A-scan and the recostructed A-scan by performing the inverse-DWT on the 98.7% 3D compressed data. High signal fidelity can be observed from Fig. 6.

B. Compression by Decimation

Another method to perform compression is by decimation. Since the signal is oversampled, there will be a lot of redundant information which can be removed by performing signal decimation. To reconstruct the signal, interpolation is performed. Signal samples that are removed through the process of decimation for compression can be recovered by interpolation method. The interpolation is achieved by padding zeros in the frequency domain representation of the signal which is explained below in detail.

Suppose a sequence x(n) is sampled periodically by keeping every J-th sample of x(n) and deleting (J-1) samples in between. This is called decimation. So we get a new sample sequence (which is the compressed signal)

$$x_j(n) = x(nJ)$$

Suppose we have N_1 samples in x(n) within a time interval $t_2 - t_1$, where t_1 is the starting instance of time and t_2 is the ending instance of time for capturing the signal. Let T_1 be the sampling period and f_{s1} be the sampling frequency ($f_{s1} = 1/T_1$).

If we increase the number of samples of $x_j(n)$ to N_2 ($N_2 > N_1$) by keeping ($t_2 - t_1$) constant, then the new sampling period T_2 will be less than T_1, because

$$N_1 \,.\, T_1 = N_2 \,.\, T_2 = t_2 - t_1 \quad (7)$$

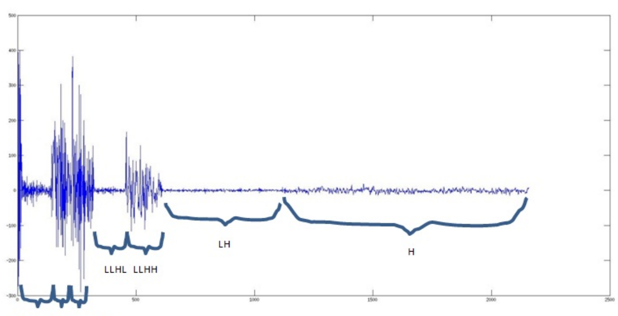

Fig. 5. DWT coefficients for one A-scan

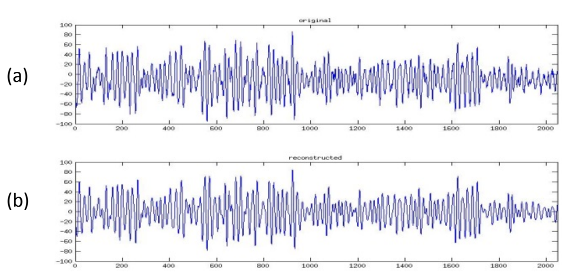

Fig. 6. Original A-scan (**top trace**) and the DWT reconstructed signal (**bottom trace**)

If we plot the frequency spectrum with N_2 samples, we can see that the frequency components beyond f_{s1} will be close to zero, provided f_{s1} and f_{s2} are higher than the Nyquist rate. Also the new sampling frequency $f_{s2} = 1/T_2$, will be greater than f_{s1}. Hence, we can write

$$f_{s2} = \frac{N_2}{N_1} \cdot f_{s1} \quad (8)$$

The ratio $\frac{N_2}{N_1}$ is called the ratio of interpolation. Maximum value of f_{s2} is the sampling frequency of the original signal (before decimation), which is 100 MHz in our experiment. By extensive evaluation of sampled ultrasonic signal using a 5 MHz broadband transducer, we have found that decimation beyond 5 samples will result in degradation in signal fidelity. Therefore, we fix f_{s1} = 20 MHz and consequently $\frac{N_2}{N_1}$ becomes 5.

To reconstruct the original signal, the following steps are performed (Fig. 7 exhibits decimation and interpolation results applied to one A-scan of 2048 samples with 80% compression).

1) Perform FFT of the decimated signal (Fig. 7b).

2) Expand by 5 times the frequency bins using Step1 by padding zeros above f_{s1} (in the high frequency region in the middle of the spectrum, see Fig. 7c).

3) Perform IFFT of the expanded spectrum which is scaled by 5 to compensate for reduced signal energy resulting from decimation. The reconstructed (or interporlated) signal is shown in Figure 7d. The interpolated result closly matches the actual signal prior to compression.

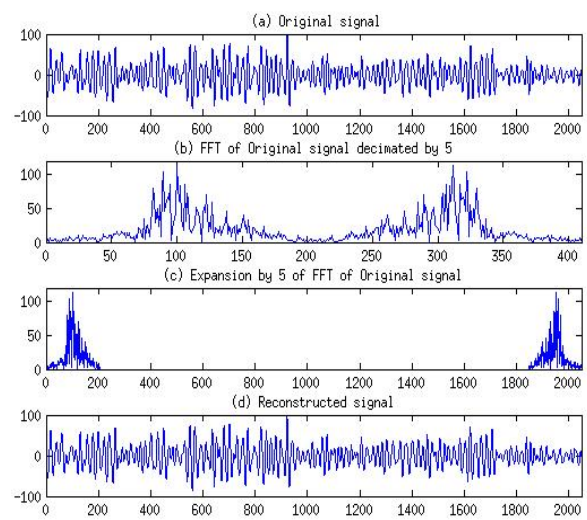

Fig. 7. Compression by decimation and reconstruction by interpolation

III. Compression performance

The ultrasonic signal compression algorithms are implemented in both hardware and software. Each of these implementation schemes has its own advantages and disadvantages. The performance of the compression algorithm based on these two schemes is explained below.

A. Hardware implementation on reconfigurable platform

The compression algorithm using 3D wavelet transform has been implemented on a reconfigurable ultrasonic system-on-chip hardware platform (RUSH). Daubechies wavelet basis is used to perform DWT in the x-direction. Haar wavelet basis is used for spatial decomposition. In the x-direction, four-level decomposition is used to generate maximum compression, while in the y and z direction, only two-level decomposition is used to improve the processing time. The RUSH platform integrates a Xilinx Virtex-5 FPGA with an embedded Microblaze processor [8]. Three processing units for compression in each directions (x, y and z) are implemented which run in parallel. There are three FIFOs to store data temporarily for each of these processing elements. Furthermore, there are three memory interface modules for accessing data to/from the external memory for each of the three axes. After processing, the result is stored into the external memory. This architecture is a modified version of the implementation given in [3]. Using this hardware implementation, a volumetric image of 128*128*2048 (8bits) samples (33 Mbytes) are compressed to 0.4 Mbytes giving a compression ratio of 98.7%. The 3D compression algorithm implemented in the RUSH platform takes around 0.5 second to compress 33 Mbytes of data into 0.4 Mbytes.

B. Software implementation on embedded processor

The ultrasonic signal compression algorithm is implemented on the Xilinx Zynq-7020 all programmable system-on-chip platform. The Zynq integrates a dual-core ARM Cortex-A9 processor based system and Xilinx programmable logic in a single device. The ARM Cortex-A9 CPUs are the main component of this system which also includes on-chip memory, external memory interfaces, and a rich set of I/O peripherals. The processors run at a frequency of 1 GHz. Each processor has its own single instruction multiple data (SIMD) media processing engine (NEON), memory management unit (MMU), and separate 32 KB level-one (L1) instruction and data caches. The Cortex-A9 processor implements the ARMv7-A architecture with full virtual memory support and can execute 32-bit ARM instructions. The NEON coprocessor's media and signal processing architecture adds instructions that target audio, video, image and speech processing and 3D graphics [11] [12]. These advanced SIMD instructions help to execute the ultrasonic signal processing algorithms at a very high rate.

The execution of 3D DWT algorithm for the volumetric image of 128*128*2048 samples takes 16 seconds on the dual core Cortex processor. The execution time can be improved by decimating the original A-scans by a factor of 4 so that we get four-time faster performance without affecting the signal fidelity. The performance can be further increased by implementing it on Zynq programmable hardware logic.

IV. Conclusion

Ultrasonic signal compression is implemented using DWT and also using decimation. DWT provides options to choose the right wavelet basis to maximize the signal fidelity and compression rate. Moreover, based on the levels of decomposition, the compression ratio can be further increased in DWT. By decimation method, the algorithm implementation becomes simpler than DWT. Nevertheless, both methods offer high signal fidelity. Thus, our study indicates that there are several alternative designs to implement the ultrasonic signal compression algorithm based on signal fidelity, computational cost and development time.

References

[1] J. Saniie "Introduction to Special Issue on Novel Embedded Systems for Ultrasonic Imaging and Signal Processing", *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 59, no. 7, pp. 1329-1332, July 2012 (invited paper).

[2] E. Oruklu, N. Jayakumar, J. Saniie "Ultrasonic signal compression using wavelet packet decomposition and adaptive thresholding", *Proceedings of IEEE Ultrasonics Symposiu,m,* pp. 171 – 175, November 2008

[3] C. Desmouliers, E. Oruklu and J. Saniie, "Adaptive 3D Ultrasonic Data Compression using Distributed Processing Engines", *Proceedings of IEEE Ultrasonics Symposium* , pp. 694 – 697, September 2009.

[4] I. Daubechies, "Ten Lectures on Wavelets'', *CBMS-NSF Regional Conference Series in Applied Math,*vol. 61, SIAM, 1992

[5] I. Daubechies, "Orthonormal bases of compactly supported wavelets'', *Communications in Pure and Applied Mathematics,*41:909, 1998

[6] G. Cui, X. Cao, Y. Wang, L. Cao, B. Huang, C. yang, "Wavelet packet Decomposition-based Fuzzy Clustering Algorithm for Gene Expression Data", *IEEE Asia Pacific Conference,* pp. 1027-1030, December 2006.

[7] R. J. Fornaris, K. W. Ferrara, B. G. Zagar, "Correlation Techniques for Ultrasonic Signal Alignment", *IEEE Ultrasonics Symposium",* vol. 2, pp. 1215-1218, October 1997.

[8] Xilinx, 2012, Platform Specification Format Reference Manual v13.4, http://www.xilinx.com/support/documentation/sw_manuals/xilinx13_4/psf_rm.pdf

[9] R. Demirli and J. Saniie, "Model-based estimation of ultrasonic echoes, part I: analysis and algorithms ", *IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control*, vol. 48, no. 3, pp. 787-802, May 2001.

[10] R. Demirli and J. Saniie, "Model-based estimation of ultrasonic echoes, part II: analysis and algorithms ", *IEEE Trans. on Ultrasonics Ferroelectrics and Frequency Control*, vol. 48, no. 3, pp. 803-811, May 2001.

[11] http://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf

[12] http://www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf