A New Lossy Compression Algorithm for Ultrasound Signals

Miguel de A. Freitas, Marcelo R. Jimenez, Henrique Benincaza and Jean P. von der Weid
Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente, 225 - 22451-040 - Rio de Janeiro, RJ - Brasil
Telephone: +5521-3527-2250 - Fax: +5521-3527-1154

***Abstract*—A new lossy compression algorithm for RF ultrasound signals (A-scan) is presented, which stores only the largest variations of the original signal. The largest variation (LAVA) algorithm is a two-step, constant bit-rate (CBR) algorithm, using only simple addition/subtraction operations, which is very suitable for FPGA implementation. The algorithm is targeted for automated ultrasonic testing (AUT) systems where a significant data reduction has been obtained while preserving the important characteristics of the signal. Comparison to other existing compression techniques is provided using the PSNR metric.**

***Index Terms*—Non-Destructive Testing (NDT), Ultrasonics, embedded systems, data compression, FPGA.**

I. Introduction

For most non-destructive testing (NDT) ultrasound systems the space required to store the acquired data is not a major concern. In many situations, it is up to the operator to decide what is important data and when to record. Even when all data must be recorded, the availability of powerful and cheap desktop computers may come into play. This is not the case of in-line inspection (ILI) tools for oil and gas pipelines, a special class of AUT systems. For ILI, the huge amount of data acquired can easily exceed both storage and throughput capabilities of the embedded electronics. The requirements of this kind of application are particularly strict: (1) they must perform very fast acquisition as, most of the time, the pipeline flow (and tool speed) cannot be reduced due to operational or economical reasons; (2) they must acquire several simultaneous channels to achieve spatial resolution; and (3) the electronics has both space and power supply availability constraints.

Traditionally the difficulties of AUT systems have been overcame by pre-processing the data in real-time in order to reduce the information to the bare minimum as, for example, when extracting only the two highest signal peaks for thickness measurement. This is known as the "gate technique" [1]. Since the original A-scan is discarded, it is not possible to do further off-line data processing to improve the measurement. For instance, if a noise spike was choosen instead of the true echo, the latter was irremediably lost.

Another data reduction technique which has been successfully employed in several AUT systems is known as the "Amplitude Time Locus Curves" (ALOK) method [1], [2]. The ALOK method is based on the identification of the half wave maximums of the RF signal. As several maximums get stored, some degree of additional data processing is still possible such as separating spikes from echoes. This method loses the signal phase information [2], so it is not possible to reproduce the correct A-scan display later or to apply some other processing techniques such as synthetic aperture focusing technique (SAFT) [1]. The ALOK method may also be used as an intermediate processing step, as in [3], where a real-time pattern recognition system performs the identification of crack-like indications.

The last data reduction technique is one that preserves the general RF signal, including both lossy and lossless compression algorithms. The most traditional RF signal compression is the "pixelisation" [1], a lossy technique which effectively reduces the amount of data as if it were acquired with a smaller sample rate. Several strategies exists for choosing which sample is to be preserved from each block of the original signal. By picking one sample at each N-samples block, a N:1 compression factor is achieved.

The pixelisation method described by Barbian [1], for instance, will choose a value within the range only if this value is greater than both first and last values of the interval, otherwise, the first value of the range is the one used. Pixelisation is very useful to store the rectified envelope of the signal. On the other hand, it is more likely to introduce phase distortions because the method might store only the most positive or negative values, therefore discarding the zero crossings.

A pixelisation-like strategy which is very effective is the Trace Compression (TC) [4]. For every N-samples block of the signal, this technique will store the maximum and minimum values while maintaining the order in which they occur (without actually saving their time positions). Some uncertainly is therefore added to their time estimation but the exact peak values are kept. Thanks to the maximum and minimum values strategy, TC will spoil zero crossings less than traditional pixelisation.

The LAVA algorithm presented here provides similar data reduction ratio and quality when compared to TC. Both are well suited for AUT applications and can provide realistic compression ratios between 10:1 and 20:1.

It should be noted that a lossless compression algorithm for the RF ultrasound signal has also been proposed using Linear Predictive Coding (LPC) [5]. This algorithm is shown to perform better than the conventional ZIP compressing tool, achieving approximately 30-55% of the original size.

Due to it's smaller compression factor when compared to a lossy algorithm, LPC is better suited for a different set of applications and will not be used as benchmark on this work[1].

II. The LAVA algorithm

The idea behind the LAVA algorithm is to keep track of the changes on the input sample values along the time by priorizing them in terms of largest absolute variations. The algorithm was designed to be mostly insensitive to the signal absolute amplitude value, in a sense that amplifying or attenuating the signal should not affect the algorithm decisions of which samples to store.

LAVA provides a CBR output, which is desirable for embedded systems as they must perform predictably. As with any other CBR lossy compression technique, an input signal richer than expected may cause loss of important signal features. However the system is meant to do its best in terms of storing something which is believed to be most relevant, instead of requiring more storage space than foreseen.

The detailed operation of the algorithm can be explained by doing an analogy with the hardware digital implementation.

The smallest logic block which constitutes the LAVA compressor can be represented electronically by (1) a register, which keeps the last value used; (2) an absolute comparator which computes the difference of a new input sample to the register value and (3) a FIFO unit. This logic block will be duplicated several times, as seen in figure 1.

Every time a new sample is provided by the analog-to-digital converter (ADC), this sample's value will be compared to the register's last value. If their difference exceeds a certain level, two events are triggered: the register is set to the new sample and the <sample, index> pair is stored on FIFO. For example, if level is set to 32 it means that, after processing the input signal, the FIFO will contain only the samples with more than 32-unit variations.

The LAVA algorithm uses several of those logic blocks wired together but with different weight each. One way of thinking about this is to label the block by the quantization bits. The 8th block will use a comparator level of 128 or 2^7 as if it was the weight of the bit 7, therefore block 7.

If the comparator of block number B is triggered (i.e., variation exceeds 2^{B-1}), every register of the block i where $i < B$ should be updated. However only the FIFO of the block B stores the <sample, index> pair. This is meant to prevent the same sample from being duplicated on several FIFOs.

This first step of the algorithm is a completely lossless operation, as long as there is enough space on FIFOs and enough blocks to represent the full ADC resolution. Data has just been reorganized in a different pairwise representation.

It is the second step that performs the lossy compression by choosing which data pairs are read back from FIFOs and outputted as the compressed stream. The LAVA algorithm performs a simple decision, poping values from the FIFOs from the highest block to the lowest until the desired output lenght is reached.

[1]A lossy extension is provided but the improvement in data rate was marginal.

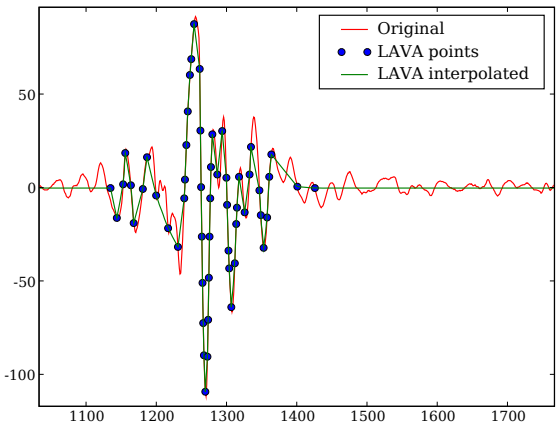


Figure 2. Detail of LAVA output and linear interpolation (30:1 data reduction).

Notice that the present algorithm is simple enough to be implemented even on lower-end FPGAs. A pipelined version of LAVA that can compress data in realtime has been tested on a Xilinx FPGA part running at 100 MHz. When a new A-scan is fed to the first step of the LAVA compressor, the data from the previous A-scan has already been saved.

Figure 2 shows a plot of a tipical A-scan data from a pulse-echo device along with a 30:1 compressed LAVA plot. Linear interpolation was used for the interpolated LAVA curve. It can be seen right away that due to the fact that LAVA gives priority to the largest signal variations, lower energy signals are wipped away. That gives the LAVA signal a "cleaner" aspect and at the same time preserving the general shape of the signal. It is not correct to say that LAVA removes the noise, because at the points chosen to represent the signal, the noise is still present. But clearly, signals with higher SNR can achieve much higher compression rates because the interesting features will be present in the largerst variation samples.

III. Comparing LAVA to other techniques

Care must be taken when comparing LAVA to other compression techniques. It is tempting to believe that reducing from a 4000 samples input to a 400-element output is the same as a 10:1 compression factor. However it is important to realize that the output from LAVA is a sequence of pairs, each containing both the sample value and the index position in the original stream. Therefore this new representation conveys more information than the original one, where the index is inferred from the sample sequence ordering.

The pair representation also requires more space for storage than the original sample. So, in order to perform a fair comparison between LAVA and other techniques, the length of the LAVA output sequence will be scaled down proportionally. For example, with a 8-bit ADC and up to 4096 samples

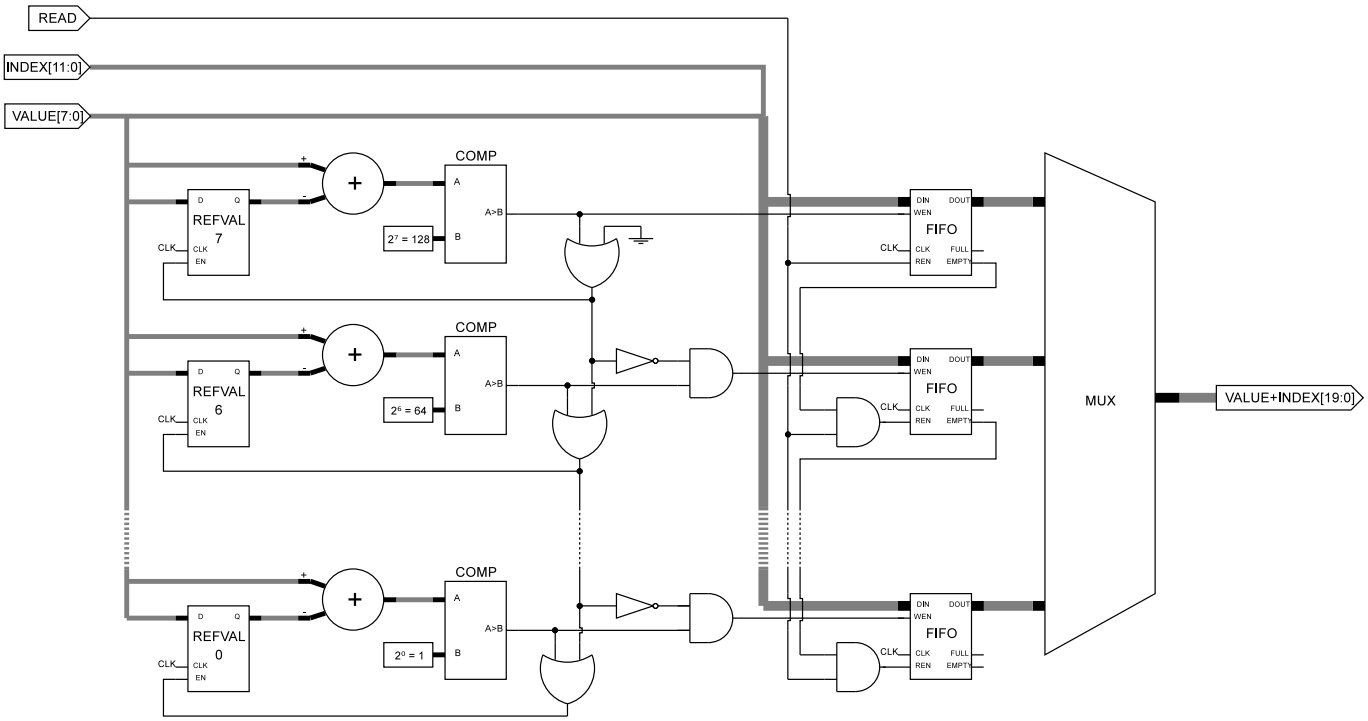


Figure 1. Block diagram of LAVA compressor.

sequence, the pair value can be stored as 8+12=20 bits=2.5 bytes. When aiming for an 10:1 data reduction from an original 4000-sample input we will set TC to produce a new 400-sample sequence. LAVA, however, is only allowed to output 2.5x less values, a total of 160 pairs.

Comparing LAVA and TC algorithm outputs is difficult and very subjective. Depending on signal's features and compression factor, one can look better than the other. It is important to realize that while TC priorizes the peak amplitude value, accepting some uncertainty on its position, LAVA will position his samples precisely but allow uncertainty at the peak value.

The figure 3 provides an example of what kind of compression is obtained from both algorithms with the same reduction rate of 16:1. LAVA clearly misses a lot of small amplitude features, while preserving frequency and shape of the signal. TC acknowledges the presence of smaller features but distorts the signal, sometimes joining two peaks into one.

A more quantitative approach for comparing the compression algorithms is to calculate the peak signal-to-noise ratio (PSNR). PSNR is commonly used as a measure of quality of signal reconstruction in image and sound compression. The typical values for PSNR in a lossy image compression range from 30 to 50dB, where higher means better fidelity as compared to original data.

Since both methods yield less samples than the original A-scan, an interpolating strategy must be choosen before PSNR can be computed. For the sake of simplicity, we have only used linear interpolation. It is reasonable to expect that using better interpolation, such as Splines, will improve PSNR figures for both LAVA and TC.

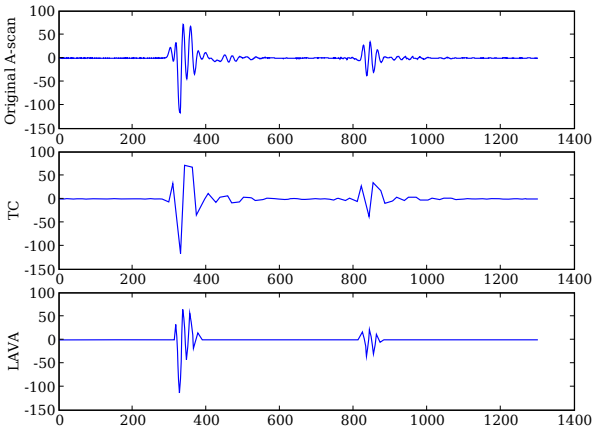


Figure 3. Comparing LAVA and Trace subjectively (16:1 data reduction).

In order to eliminate flutuations on the PSNR figure and produce unbiased results, a set of A-scans was used. This set consists of 250 A-scans acquired along an out-of-service carbon steel pipeline of 12 meters. The set represents real data from both corroded and uncorroded areas. The ultrasonic data was acquired at 100MHz, 8-bit, using a 5MHz transducer in a pulse-echo configuration. For every data reduction factor the PSNR was computed as the mean of the individual values obtained for the entire 250 A-scans set.

As can be seen on figure 4, the LAVA algorithm will consistently outperform TC in terms of PSNR values. One

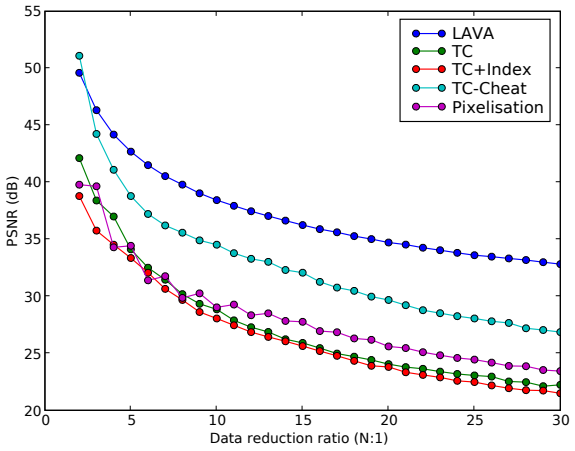


Figure 4. Using PNSR to compare compression algorithms.

possible explanation for the big difference is the uncertainty in peak position for TC, since the value of the peak is known but the position has to be estimated. In order to test this hypotesis we elaborated a modified cheating version of the TC algorithm which also stores the index, using the same pairwise representation as LAVA. This version is said to be cheating because the extra storage for the index is not taken into account in the compression ratio, so that only the difference due to the time imprecision is reflected at the PSNR value. The new compressor is shown on figure 4, and indeed, the PSNR values are better than the original TC compressor, but still inferior to LAVA by 5 to 10 dB in the plotted range. There is, however, a small part of the curve near 2:1 compression, where the cheating TC performs better than LAVA.

Figure 4 also shows that if the extra index storage is taken into consideration in the computation of the compression ratio, the new algorithm (TC+index) unfortunately performs even worst than the original TC.

IV. Conclusion

The LAVA algorithm has been shown to perform well compressing RF ultrasound signals (A-scan) for automated ultrasonic testing applications. The algorithm can be easily implemented in FPGA providing realtime data compression.

LAVA is mostly insensitive to the signal scale, meaning that amplifying or attenuating the signal should not affect the algorithm decisions of which samples to store. The algorithm is also robust to small noise as it gets naturally removed by the largest variations criteria.

Although the data reduction ratio can be set arbitrarily, realistic compression ratios while preserving good signal quality are expected in the range of 10:1 to 20:1. Our tests have demonstrated that signal reconstructed from LAVA algorithm has PSNR at least 8 dB better than Trace Compression and Pixelisation.

Additional research is needed to establish the optimum relationship between value's word length and input stream size, which increases the space needed for storing the index value. It should also be possible to achieve a smaller representation by encoding the index differentially to the previous pair.

References

[1] O. A. B. et al., *Handbook Automated Ultrasonic Testing Systems, IIW Handbook*. DVS Verlag, 2004.

[2] G. Csapo and T. Just, "Specifications of digital ultrasonic instruments in in-service inspection of nuclear power plants," *The e-Journal of Nondestructive Testing & Ultrasonics*, vol. 3, no. 5, 1998.

[3] G. Dobmann, O. A. Barbian, and H. Willems, "State of the art of inline non-destructive weld inspection of pipelines by ultrasonics," *Non-Destructive Testing*, vol. 45, no. 1, 2008.

[4] R. Mateos, J. Brizuela, M. Parrilla, A. Ibanez, and C. Fritsch, "An Ethernet-Based Ultrasound Signal Processor," *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*, pp. 1–5, 2007.

[5] A. Pesavento, V. Burow, and H. Ermert, "Compression of ultrasonic RF data," *Proceedings 1997 IEEE Ultrasonics Symposium*, vol. 2, 1997.