REAL-TIME SOUND LOCALIZATION USING FIELD-PROGRAMMABLE GATE ARRAYS

Duy Nguyen, Parham Aarabi, Ali Sheikholeslami

The Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto
10 Kings College Road
Ontario, Canada, M5S 3G4
{nguyed@ecf,parham@ecf, ali@eecg}.utoronto.ca

ABSTRACT

This paper presents a single FPGA implementation of a real-time sound localization system using two microphones. The implementation, utilizing a cross-correlation technique based on a modified version of the phase transform, successfully localizes sound sources in noisy environments with as low an SNR as 10dB. Using the same algorithm and similar hardware architecture, it is shown that up to 5 parallel systems (using 10 microphones), all real-time, can be implemented on a single FPGA while only utilizing an estimated 77mW-108mW per microphone.

1. INTRODUCTION

Real-time sound localization is required for applications such as robust speech recognition and automatic teleconferencing, where, several audio signals obtained from an array of microphones are processed concurrently [7, 5, 4].

The real-time processing of multiple audio signals is often expensive as it requires several processors. Even if a dedicated Digital Signal Processor (DSP) is employed, it often requires a large amount of power, rendering it impractical for many applications. For example, the Huge Microphone Array system developed by Brown University [12] utilized multiple DSP processors and buffers for sound localization with an average power consumption of 400mW per microphone. This is far beyond the power availability in applications such as Personal Digital Assistants and mobile phones. The best solution for such applications would be a custom-designed VLSI chip.

As a precursor of a VLSI implementation, we will illustrate the implementation of a sound localization system on a Field Programmable Gate Array (FPGA). In contrast with the prior work, which either utilized DSPs [12] or a combination of FPGAs and DSPs [10], this work implements the entire sound localization system (except the analog front end) on a single FPGA. As will be shown, the efficiency of the algorithm proposed and the hardware implementation result in an average power utilization in the 77-108mW range.

2. SOUND LOCALIZATION ALGORITHMS

While many sound localization techniques exist [7, 5, 4, 8], including signal subspace based methods such as MUSIC [11] or spatial likelihood based techniques [3, 8], the most common method is to estimate the time-delay of arrivals (TDOA) between all microphone pairs [7]. The TDOA between a single microphone pair will constrain the sound source location to a hyperboloid in three dimensions, and as a result, the intersection of multiple hyperboloids obtained from TDOA estimates from multiple microphone pairs will pinpoint the true location of the sound source.

TDOA estimation has been extensively explored in the past [9, 7]. The most common method is to utilize the generalized cross correlation (GCC) method [9]. The GCC based sound localization method is, relative to other techniques, computationally simple and efficient.

We assume that there are two microphones which receive the signals $m_1(t)$ and $m_2(t)$, respectively. These signals include noise, reverberation, and a time-delayed (with TDOA τ) version of a speech signal whose TDOA must be estimated. The most common method to estimate the TDOA is the single-segment generalized cross correlation defined below:

$$\tilde{\tau} = \arg\max_{\beta} \int_{\omega} W(\omega) M_1(\omega) \overline{M_2(\omega)} e^{-j\omega\beta} d\omega \quad (1)$$

where $\tilde{\tau}$ is an estimate for τ, $M_1(\omega)$ and $M_2(\omega)$ are the Fourier transforms of of the first and second microphone signals, respectively, and $W(\omega)$ is a cross correlation weighting function. Two different choices for $W(\omega)$ include:

$$W_{\text{PHAT}}(\omega) = \frac{1}{|M_1(\omega)||M_2(\omega)|} \quad (2)$$

$$W_{\text{UCC}}(\omega) = 1 \quad (3)$$

The PHAT weights correspond to the PHAse Transform, and are known to be effective in reverberant environments [7, 9, 4]. The UCC weights correspond to the Unfiltered Cross Correlation technique, which is simply a standard cross correlation without any weights.

The single-segment GCC for discrete-time signals can alternatively be expressed as:

$$\tilde{\tau} = \arg\max_{\beta} \sum_{k=0}^{N/2} W(k)|M_1(k)||M_2(k)|\cos(\theta(k)) \quad (4)$$

where $\theta(k) = \angle M_1(k) - \angle M_2(k) - 2\pi F_s k\beta/N$ is defined as the phase error, k is the index of the discrete fourier transforms (or, alternatively, the fast fourier transforms (FFTs)) of the signals involved, N is the total number of samples in each segment, and F_s is the sampling frequency.

Equation 4 can be viewed as a weighted reward-punish function of the phase error at different frequencies. Ideally, the phase error would be close to zero, resulting in the maximization of equation 4. The cosine phase error selector function in effect rewards lower phase errors and punishes higher phase errors. An alternative version of the equation above is to use a rectangular reward-punish function, as shown below:

$$\tilde{\tau} = \arg\max_{\beta} \sum_{k=0}^{N/2} W(k)|M_1(k)||M_2(k)|\text{rect}\left(\frac{\theta(k)}{\epsilon}\right) \quad (5)$$

where ϵ defines the width of the rectangular function (in other words, it defines the aggressivenes of the reward-punish algorithm) and $\text{rect}(t) = 1$ if $|t| < 1$ and has a value of zero otherwise.

TDOA estimation based on equation 5 has several advantages as compared to that of equation 4, including reduced implementation complexity and better performance at lower signal-to-noise ratios (SNRs). As a result, this technique was employed for FPGA implementation in this paper.

3. FPGA IMPLEMENTATION

The proposed TDOA estimation technique is employed for a single microphone pair. Each microphone undergoes amplification, bandpass filtering, and then sampling at 20kHz with 24bits per sample. The 8 most significant bits of the samples are then fed digitally to a Xilinx Virtex II 2000 (2V2000) FPGA [1] where the TDOA computation takes place. While the implementation here only consists of a single microphone pair, the extension to multiple microphone pairs is trivial and can be accomplished within the same FPGA (as will be discussed in Section 5).

The input samples are stored in two user selectable buffers (one for each channel) with sizes ranging from 256 samples to 1024 samples. The buffers were then windowed using Hanning windows, converted to 16 bit floating point representations, and stored in two Fast Fourier Transform (FFT) buffers, as shown in Figure 1. The FFT is performed in-place on each of the buffers.

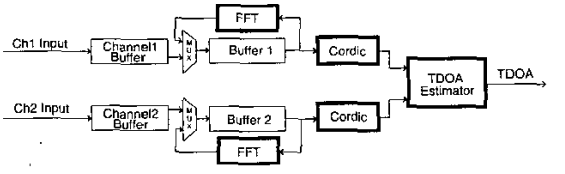

Fig. 1. The overall system. The input buffers are 1024 × 8 bits and the FFT buffers are 2 × 1024 × 16 bits.

The CORDIC algorithm is utilized to calculate the Fourier Transform sines and cosines and to convert complex numbers from real-imaginary to magnitude-phase representations [13, 6]. The CORDIC algorithm was chosen since it can perform magnitude-phase estimation quickly without significant hardware requirements.

Once the magnitude and phases of each of the two channels are calculated, the modified GCC technique of equation 5 is used to obtain a TDOA estimate, as shown in Figure 2. This TDOA estimation involves searching for the GCC maximizing τ according to equation 5. The search step size started at $-30T_s$ and ended at $30T_s$ in T_s step sizes, where $T_s = 1/F_s$ is the sampling period. Also, a value of 0.5 radians was used for ϵ. The hardware implementation uti-

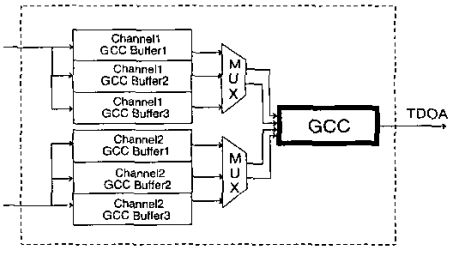

Fig. 2. The TDOA estimation block. Each GCC buffer is 2 × 256 × 16 bits.

lizes a 3-stage pipeline architecture in order to obtain real-time TDOA estimates. The first stage consists of acquiring the input samples, the second stage consists of FFT computation and conversion to magnitude-phase representation, and the final stage consists of TDOA estimation. While two GCC buffers are sufficient, three buffers are used in order to keep the results of the previous time segment. This allows for temporal smoothing which results in more accurate localizations [5].

4. REAL-TIME EXPERIMENTAL RESULTS

For the experiments in this section, the buffer size was set to 1024 bits (corresponding to approximately 50ms time segments). Two microphones were connected to the FPGA as discussed in the previous section. The first experiment involved a stationary speaker positioned in the room as shown in Figure 3. As shown in Figure 3, a single male speaker

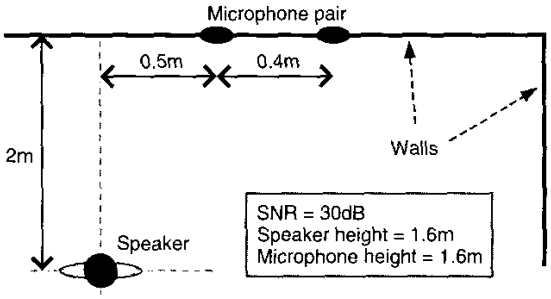

Fig. 3. Experimental setup with stationary speaker.

(whose mouth height equals the height of the microphones) speaks in a room for several minutes with two microphones placed on the walls. The microphone signals are amplified, filtered, sampled, and processed by the FPGA as discussed in Section 3.

Noise consisting of sensor noise introduced by the microphones and/or their amplification and filtering systems results in an average signal-to-noise ratio of 30dB. The UCC and the PHAT weights are used to estimate the TDOA of the speaker for every 50ms speech signal frame that is captured by the microphones. The TDOA τ of each frame is converted to a direction-of-arrival (DOA) $\phi = \arcsin\left(\frac{v\tau}{d}\right)$, where v is the speed of sound in air (approximately 345 m/s) and d is the inter-microphone distance (in this case, 0.4m).

The resulting DOA estimates are subtracted from the true DOA (which is obtained from the actual location of the speaker in the environment) in order to assess the DOA error. The resulting DOA error histogram is plotted in Figure 4. As shown, PHAT TDOA estimation results in the most accurate localization of the sound source. As a result, for all subsequent experiments, only the PHAT technique was used. In order to assess the performance of the FPGA based sound localization system for different speaker positions and noise conditions, several other experiments using the PHAT technique were performed. This time, the speaker moved from one location in the environment to another location, as shown in Figure 5. During this motion, which took approximately 1 minute, the speaker continuously spoke while facing microphones.

This experiment was performed a total of 8 times, with SNRs of 30dB, 20dB, 10dB, and 0dB, and for two different

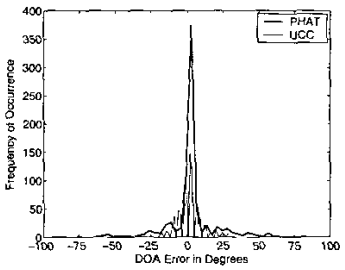

Fig. 4. Histogram of DOA errors for staionary speaker with a 30dB average SNR.

male speakers. While the 30dB average SNR condition involved no external noise sources (the noise only consisted of the inherent sensor and sensor processing noises), the other SNRs (20dB, 10dB,and 0dB) were obtained by placing a Gaussian noise source far away from the array, and adjusting its intensity to result in the desired SNR of the recorded signals. The DOA error histogram for the 30dB average

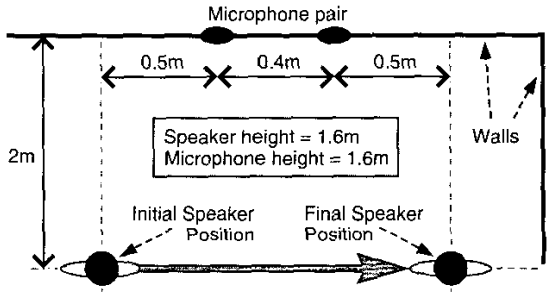

Fig. 5. Experimental setup with moving speaker.

SNR case is shown in Figure 6a. As the SNR is decreased (20dB in Figure 6b and 10dB in Figure 6c) a coherent peak is maintained near a DOA error of zero degrees, indicating that a correct sound localization is possible. However, at lower SNRs, the average error is larger (indicated by larger peaks in the histogram at non-zero errors). At 0dB, the coherent peak around zero disappears, indicating that a correct sound localization is no longer possible, as shown in Figure 6d.

5. CONCLUSIONS

A real-time sound localization system was implemented on a Xilinx Virtex II 2000 (2V2000) FPGA. It was experimentally shown that the proposed sound localization algorithm and the implementation resulted in a robust sound localization system capable of accurate localizations in SNRs as low as 10dB.

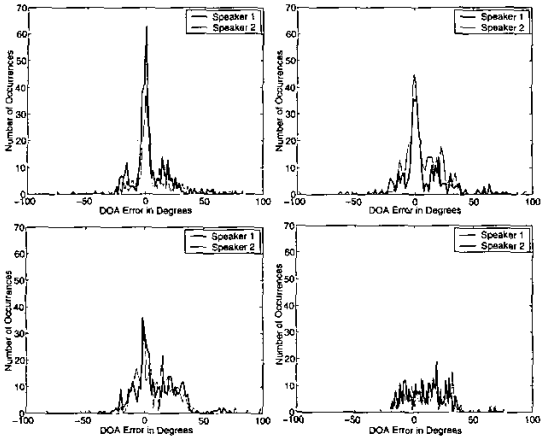

Fig. 6. Histogram of DOA errors with moving speaker with a) 30dB average SNR (top left), b) 20dB average SNR (top right), c) 10dB average SNR (bottom left), and d) 0dB average SNR (bottom right).

The standard method for implementing multi-microphone signal processing algorithms is to use Digital Signal Processors (DSPs) [12]. However, the power requirement and the need for external components (such as memory) make DSPs unsuitable for many applications that require low power consumption and a low profile. For example, using the method proposed in this paper, it is possible to have 5 to 6 parallel TDOA estimation modules (using up to 12 microphones) within a Xilinx Virtex II Pro-70 FPGA [2], running at 10MHz and consuming between 0.776W and 1.074W of estimated power. This estimate is obtained from the fact that our proposed implementation used a total of 1,192,793 logic gates and the Virtex II Pro-70 has a total of 6,000,000 gates available. By pipelining and running the same FPGA at 100MHz, it becomes possible to fit up to 50 TDOA estimation modules (using up to 100 microphones) within the same FPGA but now consuming between 7.76W and 10.74W of power. Note that the input buffers do not require extra logic gates since the Virtex II Pro-70 has enough memory for 100 input buffers.

Compared with alternative approaches (such as the Huge Microphone Array which consumed about 400mW per microphone [12]), the average power consumption of the proposed FPGA system is far less (about 77-108mW per microphone). The only cause for concern regarding FPGAs is the relative cost compared to other approaches. However, for most commercial applications such as sound localization on handheld computers, custom VLSI circuits can be fabricated which would have a reduced cost (when produced on a commercial scale), low power consumption, and more processing capability.

6. ACKNOWLEDGEMENTS

We would like to thank Francis Leung for his contributions to the experiments carried out in this paper.

7. REFERENCES

[1] Xilinx Virtex II 2000 (2V2000) datasheet. This document can be found at *www.xilinx.com/products/tables/fpga.htm#v2*.

[2] Xilinx Virtex II Pro power consumption datasheet. This document can be found at *www.xilinx.com/support/techsup/powerest/index.htm*.

[3] P. Aarabi. The integration of distributed microphone arrays. In *Proceedings of 4th International Conference on Information Fusion*, July 2001.

[4] P. Aarabi and A. Mahdavi. The relation between speech segment selectivity and time-delay estimation accuracy. In *Proceedings of ICASSP*, May 2002.

[5] P. Aarabi and S. Zaky. Robust sound localization using multi-source audiovisual information fusion. *Information Fusion*, 3:2:209–223, September 2001.

[6] R. Andraka. A survey of CORDIC algorithms for FPGAs. In *FPGA'98*, 1998.

[7] M.S. Brandstein and H. Silverman. A robust method for speech signal time-delay estimation in reverberant rooms. In *Proceedings of ICASSP*, May 1997.

[8] J. DiBiase, H. Silverman, and M. Brandstein. Robust localization in reverberant rooms. *M.S. Brandstein and D.B. Ward (eds.), Microphone Arrays:Signal Processing Techniques and Applications*, 2001.

[9] C. H. Knapp and G. Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on ASSP*, 24(4):320–327, August 1976.

[10] M. Ponca and C. Schauer. FPGA implementation of a spike-based sound localization system. In *5th International Conference on Artificial Neural Networks and Genetic Algorithms*, 2001.

[11] R. O. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, AP-34:276:280, March 1986.

[12] H. F. Silverman, W. R. Patterson III, and J. L. Flanagan. The Huge Microphone Array (HMA). *Brown University Technical Report*, May 1996.

[13] J.E. Volder. The CORDIC trigonometric computing technique. *IRE Transactions*, EC-8:330:334, 1959.