ICIC Express Letters
Volume **10**, Number **12**, December **2016**

ICIC International ©2016 ISSN 1881-803X
pp. **2949–2955**

FRAME-BY-FRAME SPEECH SIGNAL PROCESSING FOR FIELD PROGRAMMABLE GATE ARRAY DEVICES

Masashi Nakayama[1], Naoki Shigekawa[2] and Takashi Yokouchi[3]

[1]Graduate School of Information Sciences
Hiroshima City University
3-4-1 Ozuka-higashi, Asaminami, Hiroshima 731-3194, Japan
masashi@hiroshima-cu.ac.jp

[2]Graduate School of Engineering
University of Fukui
9-1 Bunkyo, Fukui 910-8507, Japan

[3]National Institute of Technology
Kagawa College
551 Khoda, Takuma, Mitoyo, Kagawa 769-1192, Japan
yokouchi@cn.kagawa-nct.ac.jp

Received May 2016; accepted August 2016

Abstract. *This paper proposes the frame-by-frame speech signal processing technique that is suitable for implementation in Field Programmable Gate Array (FPGA) device. Several applications are proposed, including voice conversion system that needs real-time signal processing and speech recognition for each frame. Because high speed processing needs to pre-process speech including voice changer and so on, the authors propose algorithms to implement for Voice Activity Detection (VAD) on FPGA. The algorithms are customized for the VAD algorithm using second order autocorrelation function because FPGA has little resources for calculation. These methods were implemented and tested on an FPGA emulator to demonstrate the VAD algorithm for speech in both quiet and noisy environments.*

Keywords: Speech signal processing, Voice Activity Detection (VAD), Speech recognition, Autocorrelation function, Field Programmable Gate Array (FPGA)

1. **Introduction.** Speech interfaces, including speech recognition and voice synthesis systems, are now introduced into many applications, such as cellphones, computing, and vehicle navigation. Speech recognition is calculation intensive and requires many processing times when state-of-the-art statistical decoding algorithms are used, such as Hidden Markov Models (HMMs), Deep Neural Networks (DNNs) or Gaussian Mixture Models (GMMs) for discrimination and recognition processing [1,2]. In addition, speech recognition systems are not able to process data instantaneously that consists only of a few frames due to insufficient time and resources. Consequently, these algorithms are not applied to work on laptop or workstation directly. For these reasons, fast, lightweight, and low energy are needed to enable speech interfaces for wearable and portable devices. Contribution to performing on hardware using Digital Signal Processors (DSPs) and Field Programmable Gate Arrays (FPGAs) is ongoing because hardware-based processing is faster than software-based processing. Several studies have attempted to develop a speaker discrimination decoder for small vocabulary on microcomputer and DSPs [3-5]. Other researchers have implemented these decoders using conventional statistical decoding techniques on hardware, including FPGA devices [6,7]. One group showed a speech recognition algorithm that used HMM to a microcomputer, which has semi-continuous distributions as Gaussian model [8].

For faster processing, it is a better solution that hardware implementation uses logical gates. In this paper, the authors propose the algorithm for sound quality improvement using a frame-by-frame processing signal estimation technique [9-11]. The method recognizes vowels, consonants, and unvoiced sections in each frame. However, it must be implemented in FPGAs because conventional processing techniques used in microcomputers and DSPs are too slow. On the other hand, conventional algorithms use processing techniques that are too complex for FPGAs, so we propose a new hardware decoding method of Voice Activity Detection (VAD) and speech recognition for implementation in FPGA devices. Especially, this paper focused on VAD algorithm implemented for FPGA device. Therefore, the fundamentals of speech signal processing in FPGAs will appear and be implemented on an FPGA emulator here.

This paper is constructed as follows. Section 2 shows the overview of the proposed system for implementing frame-by-frame speech signal processing on an FPGA device. Section 3 shows VAD using a second order autocorrelation function for the preprocessing of speech recognition. Finally, Section 4 discusses the conclusions and future work derived from this research.

2. Frame-by-Frame Speech Signal Processing on FPGA Devices.

Figure 1 shows the overview of the proposed system on an FPGA device. There are two main processing stages: signal processing and speech recognition. First, the system discriminates speech and no-speech frames by the VAD using second order autocorrelation function with sound quality improvement. Next, the detected frames as speeches are recognized vowels referred by templates of parameters.

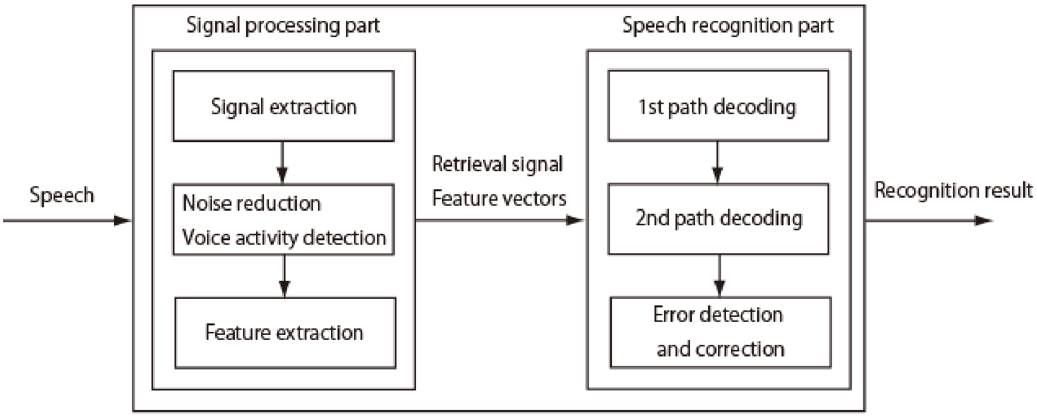


Figure 1. Frame-by-frame speech signal processing and recognition in an FPGA

During the first stage of speech processing, the system extracts framed data from speech signal. Then, each frame is also analyzed on the consonants or noise, and assigned a frame status. In general, autocorrelation function analyzes and detects the periodicity of sound based on the fundamental frequency of speech. In this research, second order autocorrelation function was employed because one of the authors determined that this order of function was best able to emphasize the robustness of periodicity detection. The effective duration of the autocorrelation envelope τ_e is used because its parameters are expected to represent the one-dimensional parameters of slope angle and/or decreasing magnitude level [12]. However, the parameters for this research have been modified in order to facilitate implementation in an FPGA device. In addition, noise reduction processing can also be optionally applied to the frame, and then combined with the results of the discrimination between speech nor no-speech status. During the second stage of speech recognition, the system recognizes vowels or consonants referred templates and feature parameters from sounds. This paper describes focus on the 1st stage at the processing.

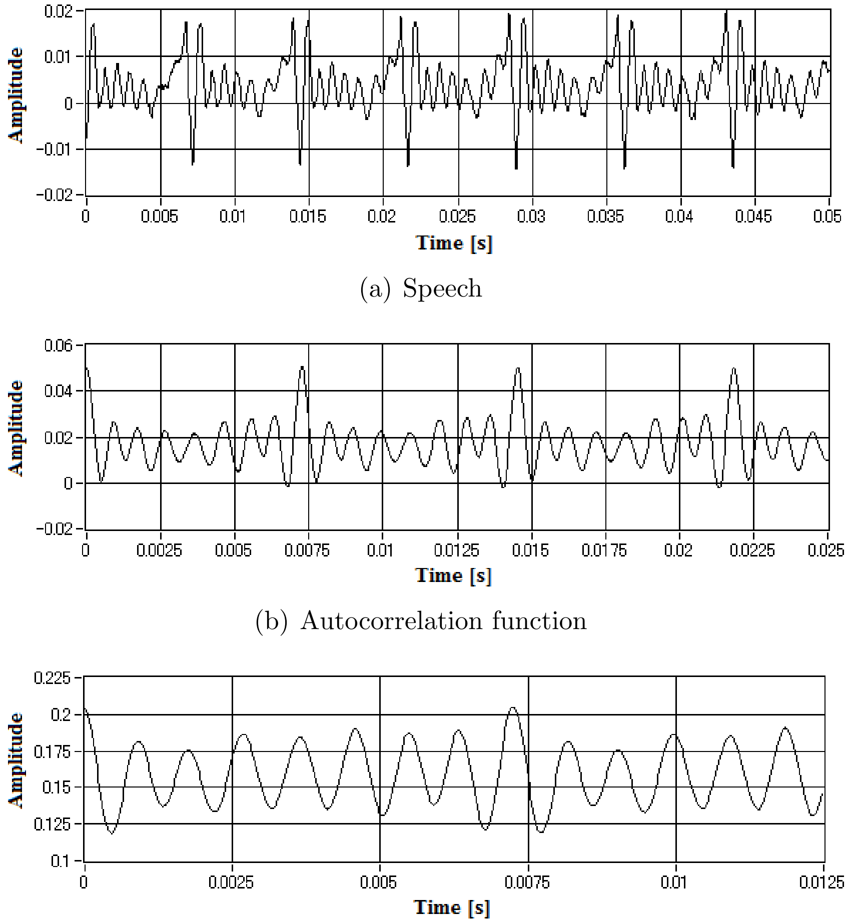
3. Voice Activity Detection Using Second-Order Autocorrelation.

3.1. Robustness of the second-order autocorrelation. The calculation resources in an FPGA are limited. Therefore, the VAD algorithm and recognition methods must depend on only a few calculations. In this section, VAD using a second order autocorrelation function is proposed. Autocorrelation is one of the most well-known analysis methods used for evaluating the periodicity of a signal [13]. The conventional autocorrelation $R(j)$ can be calculated using the following equation:

$$R(j) = \frac{1}{N}\sum_{i=1}^{N} x(i) \cdot x(i+j) \tag{1}$$

where $x(i)$ is the speech signal, i is the time index, and j is the integral index. The value of $R(j)$ increases if the sound has a periodic signal, and decreases when there is little periodicity. This research employs the LabVIEW development environment, which is a commonly used development environments for FPGA devices [14]. The development environment is composed of Graphical User Interface (GUI) and compilers for FPGA circuits.

Figures 2(a) to 2(c) show speech of the vowel /a/ in its original form and autocorrelation functions. The vowel was uttered speech by a 20-year-old male in quiet conditions. Figure 2(a) shows the speech, Figure 2(b) shows the autocorrelation of the speech, and Figure 2(c) shows second order autocorrelation of the speech. Both Figures 2(b) and 2(c) confirm the periodicities of the speech, and the waveform in Figure 2(c) has a stronger periodicity.



(a) Speech

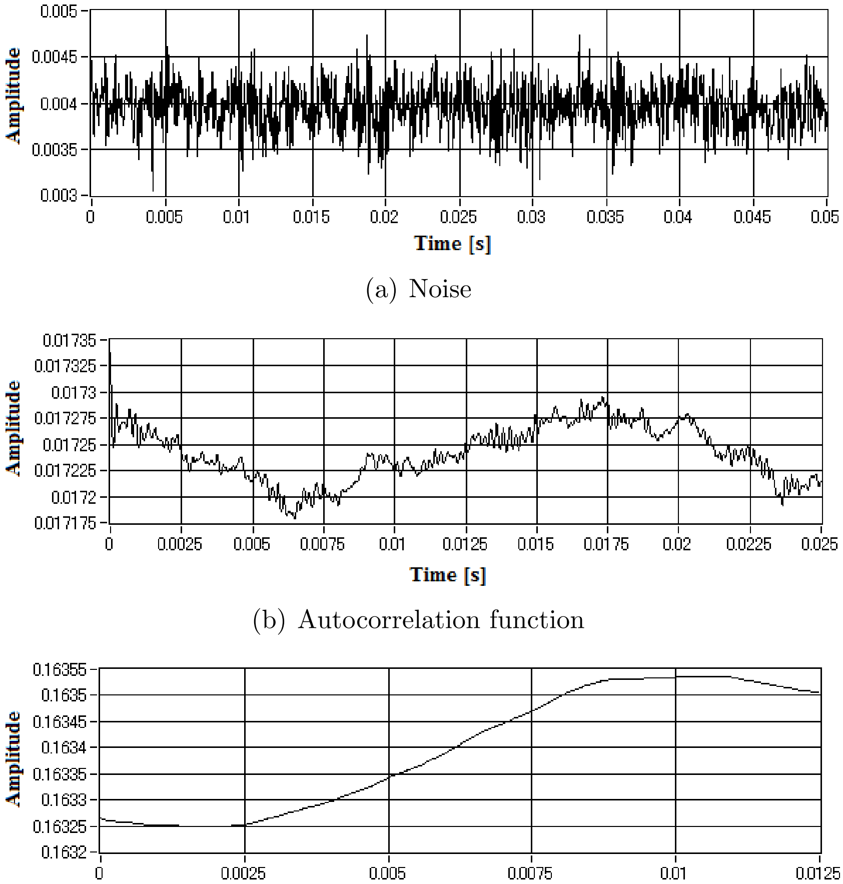
(b) Autocorrelation function

(c) 2nd-order autocorrelation function

FIGURE 2. Autocorrelation functions of speech

This periodicity detection is more robust than when the conventional autocorrelation method is employed.

By contrast, Figures 3(a) to 3(c) show the original sound and autocorrelation functions when one of the samples is an aperiodic sound, such as noise: Figure 3(a) shows the unstable noise, Figure 3(b) shows the autocorrelation of this noise, and Figure 3(c) shows the second order autocorrelation. The curve in Figure 3(c) shows low amplitude when compared to Figure 2(c) because the original waveform did not exhibit any periodicity. Based on these results, the second order autocorrelation function is clearly suitable for speech detection compared to conventional autocorrelation function.



(a) Noise

(b) Autocorrelation function

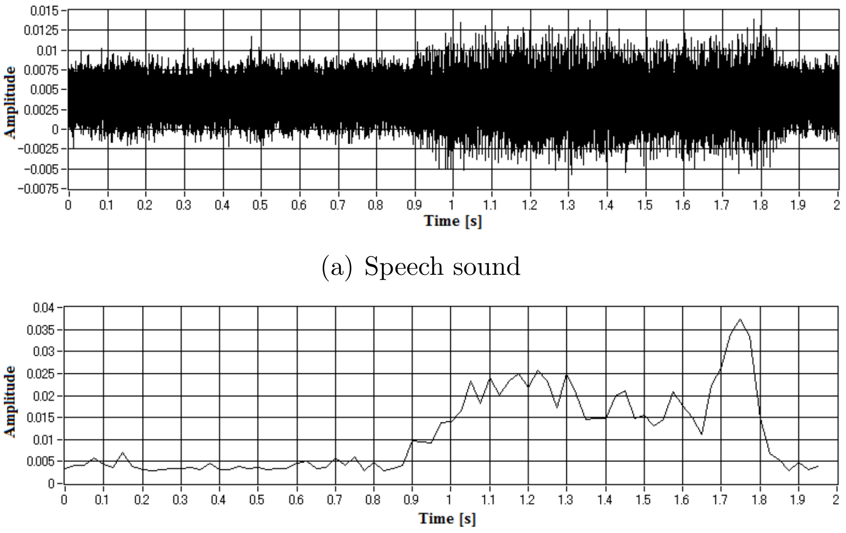
(c) 2nd-order autocorrelation function

Figure 3. Autocorrelation functions of noise

3.2. **VAD for isolated vowels.** Section 3.1 describes the robustness of the second order autocorrelation. It should be noted that the autocorrelation cannot be used directly for detecting speech including vowels. Instead, it is necessary to first compute the second order autocorrelation value, and then convert it into the estimated value of the voiced section $A(l)$, as in the following equation:

$$A(l) = \sum_{k=0}^{K-1} |R'(k+1) - R'(k)| \tag{2}$$

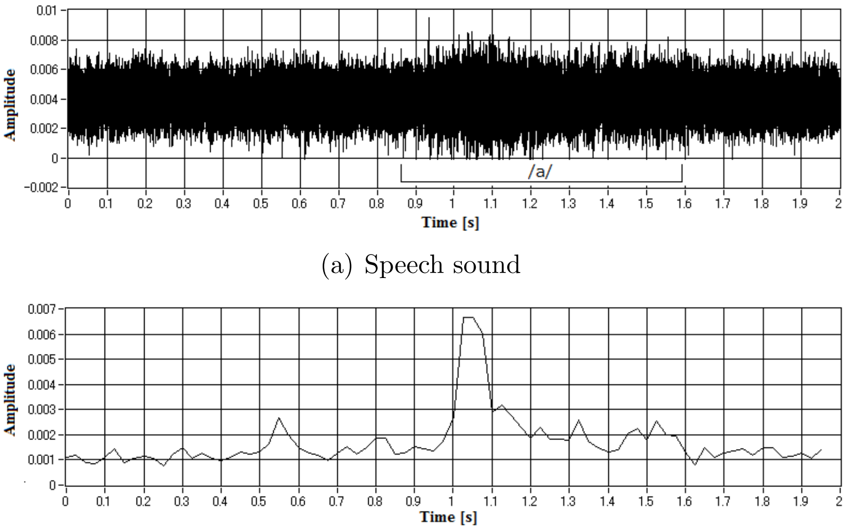
Generally, speech sounds are composed of a combination of consonants and vowels, and the vowels have a periodicity caused by the pitch generated by the vocal chords. For this reason, Equation (2) has a high value except during the utterance of consonants and silence; hence, it is easy to detect voiced sections. In contrast, the value is low when



(a) Speech sound

(b) VAD result

Figure 4. VAD of /a/ with high SNR



(a) Speech sound

(b) VAD result

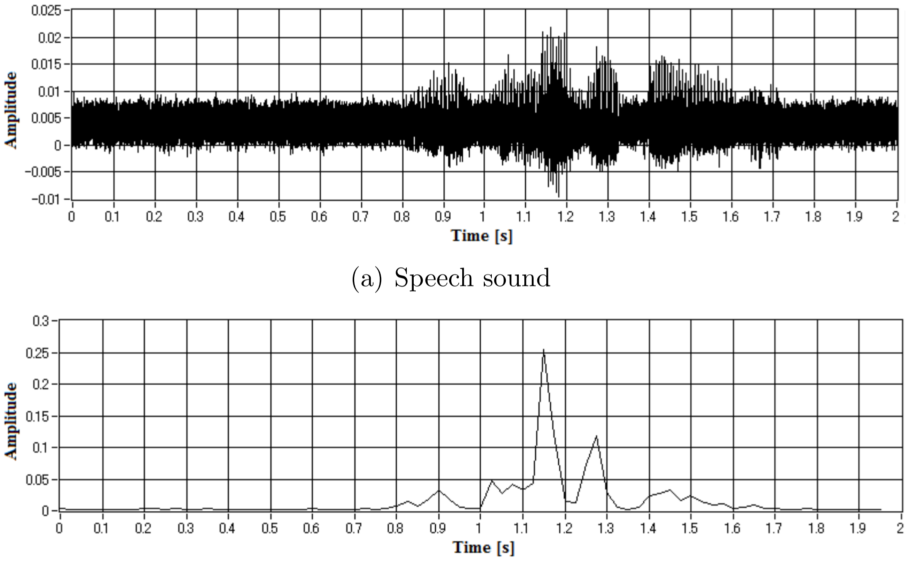
Figure 5. VAD of /a/ with low SNR

the speech is unvoiced sound or there is stationary noise such as periodic sound which do not work correctly. There is a weak point in this approach because the value is high when the background noise has periodicity. However, the benefit is that this algorithm does not require any initial settings, unlike those used for parameter estimations. So the most aggressive points are no initial settings for working VAD because the VAD algorithm operates correctly when there is no implementation for settings. As a result, this detection method is quite robust in most situations. The detection performance depends on the Signal-to-Noise Ratio (SNR). If the amplitude of the signal is larger than that of the noise, the SNR is high; otherwise, the SNR is low. In this evaluation, noisy environments with high and low SNRs were experimented. Figure 4(a) shows the vowel /a/ with a high SNR, and Figure 4(b) shows its VAD. From Figure 4(b), it is clear that the VAD algorithm performs correctly because the utterance was detected between approximately 0.9 and 1.8 s. Figure 5(a) shows a vowel /a/ with a low SNR, and Figure 5(b) shows its

VAD. As in the high SNR case, the VAD algorithm correctly determines the utterance to be between 0.9 to 1.8 s for a low SNR signal.

3.3. **VAD for sentence units.** To evaluate the overall performance of the VAD algorithm, the ATR 503 sentence unit was selected from among a number of example sentences intended for speech research in Japanese. For the experiment, a recording was made of a 20-year-old male speaking sentence A01 from ATR 503 once [15].

Figure 6(a) shows the sentence when vocalized in a high-SNR environment, and Figure 6(b) shows the VAD result. From Figure 6(b), it is clear that the vowel sections generated high VAD values and the consonants generated low values, while the noise sections are much lower than either of the voiced sections. These results show that it is possible in practice to use VAD for detecting the spoken sentence.



(a) Speech sound

(b) VAD result

Figure 6. VAD of Japanese sentence

4. **Conclusions and Future Work.** This paper proposed and evaluated the frame-by-frame speech recognition method that is suitable for implementation on FPGAs. Algorithms implemented in FPGAs are expected to be faster than calculations performed by conventional microcomputers and DSPs. However, the tradeoff is that FPGAs have limited resources. The system was demonstrated on an FPGA emulator, and showed the effectiveness and robustness of the VAD. Along with this proposed method, it is expecting that the sound quality improvement algorithm can be implemented on an actual system to ensure clear sound estimation [9]. As a direction for future work, the authors plan to implement such an algorithm on FPGA.

REFERENCES

[1] J. Benesty, M. Sondhi and Y. Huang, *Springer Handbook of Speech Processing*, Springer, 2008.

[2] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*, Springer, 2015.

[3] K. Kokubo, N. Hataoka, T. Lee, T. Kawahara and K. Shikano, Computational reduction of continuous speech recognition software "Julius" on super microprocessor, *Journal of Information Processing*, vol.50, pp.2597-2606, 2009 (in Japanese).

[4] C. G. Concejero, V. Rodellar, A. A. Marquina, E. M. de Icaya and P. Gomez-Vilda, Designing an independent speaker isolated speech recognition system on an FPGA, *Research in Microelectronics and Electronics*, pp.81-84, 2006.

[5] S. Nedevischi, R. K. Patra and E. A. Brewer, Hardware speech recognition for user interfaces in low cost, low power devices, *Proc. of the 42nd Design Automation Conference*, pp.684-689, 2006.

[6] K. Okamoto, H. Tamukoh and M. Sekine, Sound preprocessing circuit by consonant and vowel recognition system, *IEICE Technical Report*, VLD2011-93 (CPSY2011-56, RECONF2011-52), pp.13-18, 2012 (in Japanese).

[7] S. J. Melnikoff, S. F. Quigley and M. J. Russell, Implementing a simple continuous speech recognition system on an FPGA, *Proc. of the 10th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, pp.275-276, 2002.

[8] S. J. Melnikoff, S. F. Quigley and M. J. Russell, Speech recognition on an FPGA using continuous hidden Markov models, *Proc. of the 12th International Conference on Field-Programmable Logic and Applications*, pp.202-211, 2002.

[9] M. Nakayama, *Japan Patent JP2011-84323 (JP2012-220607A)*, 2011.

[10] M. Nakayama, N. Shigekawa and T. Yokouchi, Hardware speech recognition system for processing and recognition at moment, *IEICE Technical Report*, EA2010-99 (2010-12), 2010 (in Japanese).

[11] M. Nakayama, N. Shigekawa, T. Yokouchi and S. Ishimitsu, Frame-by-frame speech recognition as hardware decoding on FPGA devices, *The 9th International Conference on Sensing Technology (ICST 2015)*, Auckland, New Zealand, pp.785-788, 2015.

[12] K. Kato, K. Fujii, K. Kawai, Y. Ando and T. Yano, Blending vocal music with a given sound field due to the characteristics of the running autocorrelation function of singing voices, *J. Acoust. Soc. Am.*, vol.115, pp.2437, 2004.

[13] L. R. Rabiner, On the use of autocorrelation analysis for pitch detection, *IEEE Trans. Sig. Process.*, vol.25, pp.24-33, 1977.

[14] LabVIEW, *National Instruments Corporation*, http://www.ni.com/labview/, 2016.

[15] ATR 503 Sentences, *Speech Resources Consortium (Japanese)*, http://research.nii.ac.jp/src/ATR503.html, 2016.