Data Aggregation Protocol for Multiple Sound Sources Acquisition with Microphone Array Network

Shintaro Izumi, Hiroki Noguchi, Tomoya Takagi, Koji Kugata, Shimpei Soda,
Masahiko Yoshimoto, and Hiroshi Kawaguchi
Graduate School of System Informatics
Kobe University
Kobe, Japan
shin@cs28.cs.kobe-u.ac.jp

***Abstract*—In this paper, we propose a microphone array network that realizes ubiquitous sound acquisition for multiple sound sources. Several nodes with 16 microphones are connected to form a huge sound acquisition system, which carries out voice activity detection (VAD), sound source localization, and enhancement. The three operations are distributed among nodes with multi-hop data communication. Using the distributed network, we produce a low-traffic data-intensive array network. In the sound-source enhancement, we combine the delay-and-sum beam-forming algorithm with network data aggregation. The prototype of microphone array node is implemented in SUZAKU FPGA boards, which demonstrates a real-time multiple-sound-source enhancement operation.**

Keywords—microphone array; data aggregation; sound sources acquisition; FPGA implementation

I. INTRODUCTION

In recent years, digital human interfaces have been developed for living spaces, medical centers, robotics, and automobiles. Future applications will enable one person to control thousands of microprocessors without consciousness of their existence. Some face and speech recognition systems are in practical use, but most systems operate only in constrained environments or strictly defined installation conditions, with parameters such as an angle or distance to a device. Users must confront a camera in a face recognition system; a microphone must be near a mouth in a speech recognition system. For most people, these constraints are inconvenient for daily life. Therefore, various intelligent ubiquitous sensor systems have been developed as new human interfaces. In future applications, numerous cameras and microphones will be emplaced on walls and roofs of living spaces. They will obtain visual information and speech data automatically and support absolutely hands-free systems. As described herein, we specifically examine speech signal processing as a ubiquitous sensor system because a speech interface is a fundamental mode of human communication; moreover, a speech interface has a much broader range of applications.

Recent improvements in information processing technology have produced real-time sound-processing systems using microphone arrays [1]. One application is a meeting system with a 128-ch square microphone array [2], which captures speech data from every microphone: The microphone array processes signal recordings and also performs noise reduction, sound-source separation, speech recognition, speaker identification, and other tasks.

A microphone array can localize sound sources and separate multiple sources using spatial information of the acquired sounds. Huge microphone arrays have been widely investigated: arrays have been built at Tokyo University of Science (128 ch) [2], the University of Electro-Communication (156 ch) [3], Brown University and Rutgers University (512 ch) [4], and the Massachusetts Institute of Technology (1,020 ch) [5]. Nevertheless, their practical use is confounded by difficult problems of increasing computation, power consumption, and network cost, particularly in terms of sound-data acquisition. The salient difficulty of conventional microphone array systems is that all microphones are connected to a single base station (high-performance sound server) with large-scale multi-channel sound recording devices. In conventional systems, concentrative connection of numerous microphones engenders heavy network traffic. The computational effort increases as the number of microphones increases. If more than 1,000 microphones are used to collect the data, then the signal-noise ratio (SNR) can be improved remarkably [5], but the network traffic and computational amounts skyrocket. To reduce the increased network traffic and computational power of a microphone array system and to satisfy recent demands for ubiquitous sound acquisition, it is necessary to realize a large sound-processing system covering a wide-ranging human environment at low power.

To implement a microphone array as a realistic ubiquitous sound acquisition system with scalability, we propose division of the huge array into sub-arrays to produce a multi-hop network: an intelligent microphone array network [6–9]. The sub-array nodes with some microphones can be set up on the walls and ceiling of a room. Reducing the amount of transmission can be accomplished after introducing multi-hop networking. Each relay node on a routing path must store all temporal multi-channel sound data that the node receives, but cannot send all the data. This function would demand a large buffer memory and large total power dissipation in the system. Therefore, some breakthrough network solution is necessary to reduce the network traffic. Herein, we describe how our

978-1-4577-0636-3 /11/$26.00 ©2011 IEEE

microphone array system solves the problems described above and handles multiple sound sources. Multi-hop networking, specific data aggregation, and distributed processing are novel concepts unlike conventional microphone array systems. The performance can be improved easily by increasing the node number, but communication among nodes does not increase much in our system.

The proposed distributed sound acquisition system is presented in Section II. The prototype FPGA implementation and evaluation are explained in Section III, where we discuss the performance and accuracy of the proposed system. Section IV concludes this paper.

II. Intelligent Microphone Array Network

This section describes the proposed microphone array system.

A. System Design

Fig. 1 presents a brief description of the microphone array network system and a functional block diagram of a sub-array node. Sixteen-microphone inputs are digitized using A/D converters; the sound information is stored in SRAM. Then, the information is used for sound source localization and sound source separation. The sound-processing unit including them is activated by the power manager and voice activity detection (VAD) module to conserve standby power: The sound processing unit is turned off if no sound exists around the microphone array. Power management is fundamentally necessary because enormous microphones waste much power when they are not in use.

Fig. 2 portrays a flow chart of our system. The salient features of the system are: 1) low-power voice activity detection to activate the entire node, 2) sound-source localization to locate sound sources, and 3) sound-source separation to reduce the sound noise level. The sub-array nodes are connected to support their mutual communication. Therefore, the sound gained by each node can be gathered to improve the sound source's SNR further. The system constitutes a huge microphone array through interaction with surrounding nodes. Therefore, computations can be distributed among nodes. The system has scalability in terms of the number of microphones. Each node preprocesses acquired sound data.

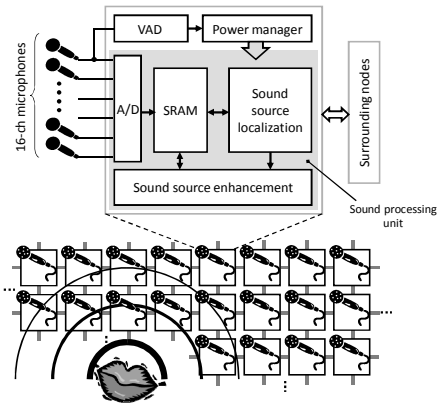

Figure 1. Intelligent microphone array network and sub-array node.

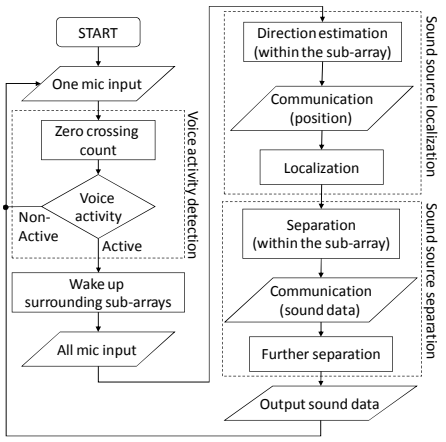

Figure 2. Flow chart of microphone array network system.

B. Voice Activity Detection (VAD)

A microphone array network consists of numerous microphones, whose power consumption would easily become prohibitive. Our intelligent microphone array system must operate with a limited energy source to save power to the greatest extent possible. Sound processing that conserves power is effective because the sound processing unit and microphone amplifiers consume some power even when the surroundings are silent.

In our previous work, we proposed a low-power VAD hardware implementation to reduce the standby power of sub-arrays [6]. This custom hardware uses a zero-crossing algorithm for the VAD. Fig. 3 portrays the zero-crossing algorithm. The zero crossing is the first intersection between an input signal and an offset line after the signal crosses the trigger line: the high trigger line or low trigger line. Between a speech signal and non-speech signal, the appearance ratios of this zero crossing differ. The zero-crossing VAD detects this difference and outputs the beginning point and the end point of a speech segment. For the zero-crossing VAD to detect speech, the only requirement is catching the crossing over the trigger line and the offset line. A detailed speech signal is unnecessary. Consequently, the sampling frequency and the number of bits can be reduced.

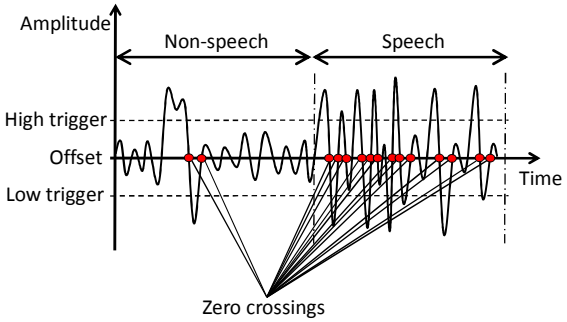

Figure 3. Zero-crossing VAD.

In our VAD, the sampling frequency can be reduced to 2 kHz and the number of bits per sample can be set to 10 bits. A single microphone is sufficient to detect a signal. The remaining 15 microphones are turned off as well. These values

are sufficient to detect human speech, in which case only 3.49 μW is dissipated on a 0.18-μm CMOS process.

By separating the low-power VAD module from the sound processing unit, it can turn off the sound processing unit using the power manager. Furthermore, not all VAD modules in all nodes need operate. The VAD modules are merely activated in a limited number of nodes in the system.

Once the VAD module detects a speech signal, the main signal processor begins to run and the sampling frequency and the number of bits are increased to sufficient values. These parameters, which determine the analog to digital converter (ADC) specifications, can be altered depending on the specific applications that are integrated on the system.

C. *Distributed Sound Acquisition*

To obtain high-SNR speech data, two types of major sound-source enhancing methods have been proposed; 1) geometric techniques, which use position information, and 2) statistical techniques, which use no position information. For the proposed system, we chose a delay-and-sum beam-forming algorithm [10] (Fig. 4), which is categorized as a geometric method, because we assumed that node positions in the network are known. This method produces less distortion than statistical techniques do. Fortunately, it requires only a small amount of computation and it is applicable to distributed processing easily. The key point to gather sound data from distributed nodes is to align sounds' phases among neighboring nodes; the phase mismatch (= time delay) is caused by the differences in distance from a sound source to each node.

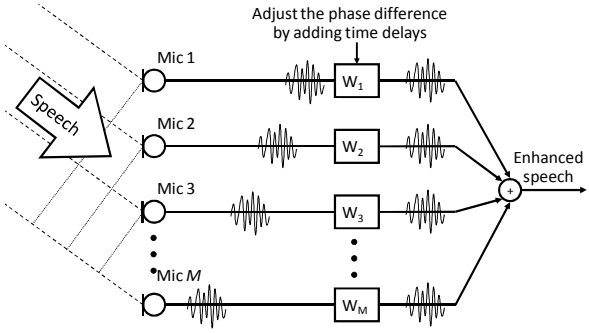

Figure 4. Delay-and-sum beam-forming.

We introduce two-layer algorithm to realize the distributed delay-and-sum beam-forming as illustrated in Fig. 5. In the local layer, each node collects 16-ch sounds that have "local" delays from the origin of the node; then an enhanced single sound is obtained within the node using the basic delay-and-sum algorithm.

Next, the enhanced sound with a constant "global" delay that can be calculated by the sun-array position is transmitted to a neighboring node in the global layer; finally, it aggregates into a high-SNR sound. A sound packet contains a timestamp and 64-sample sound data: The time stamp is given by $T_{\text{Packet}} = T_{\text{REC}} - D_{\text{Sender}}$, where T_{REC} denotes a timer value in the sender node when the sound data in the packet were recorded, and D_{Sender} denotes the global delay at the origin of sender node. In a receiver node, the received time stamp is adjusted by adding its global delay (D_{Receiver}) to T_{Packet}, and then the sound data are summed up in the delay-and-sum manner. Consequently, a high-SNR speech data can be acquired at a base station, although each node transmits only single channel sound data.

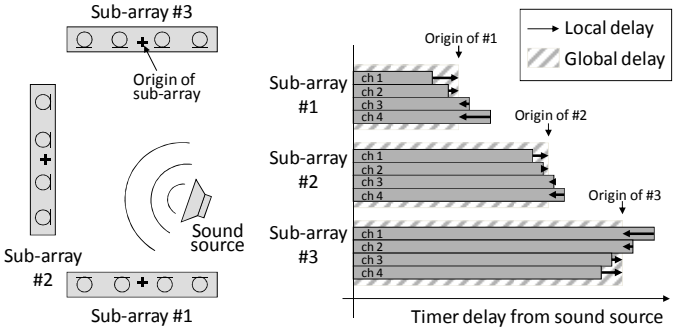

Figure 5. Distributed delay-and-sum
(# of sub channels is four for simplisity).

D. *Sound Source Localization*

However, without a precise sound source coordinate, the delay-and-sum beam-forming method does not operate effectively. For this reason, a basic sound source localization algorithm with a high degree of accuracy is important. To achieve highly accurate sound source localization, we use a hierarchical sound-source localization method [7] based on the multiple signal classification (MUSIC) algorithm [11].

The authors have proposed a calibration method with a three-dimensional coordinate of the sound source, as presented briefly in Fig. 6 [7]. First, the maximum $P(\theta,\varphi)$ and corresponding θ and φ are calculated on each node using the MUSIC algorithm. We alternatively adopt the shortest line segment connecting two lines because we can usually find no exact intersection in the three-dimensional space. We presume a point that divides the shortest line segment by the ratios of $P(\theta,\varphi)$ as an intersection. The sound source is localized by calculating the center of gravity as well, with the obtained intersections.

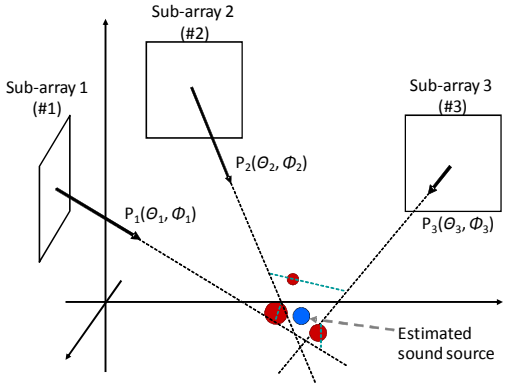

Figure 6. Sound source localization.

In a multiple sound source environment, the source positions are estimated by clustering. Since the MUSIC algorithm can estimate the direction of the loudest source, the intersections are spatially grouped around the sound sources. Therefore, the multiple source localization is achievable by clustering intersections and by calculating a center of gravity on a group-by-group basis.

E. *Time Syncronization*

Regarding sound-source enhancement, we use delay-and-sum beam-forming both within a node and among nodes. Then, the time accuracy between nodes strongly affects the final SNR of the sound source collected with the network. Because each node has its own oscillator and its own timer count, the time synchronization method between nodes is required.

To synchronize sub-arrays, we use a FTSP (Flooding Time Synchronization Protocol [12]) in consideration of the balance of hardware and power overhead. Fig. 7 shows the synchronization method of the FTSP algorithm. First, the synchronous packet is flooding from the root node. At this time, the time stamp of T_1 is send to the receiver node. Next, the receiver node records the time stamp of T_2 at the end of the synchronous packet. Then $T_2 - T_1$ means the sum of the propagation delay and the difference of the timers between a sender and a receiver. However, note that, until the synchronous packet is recognized by a system, there is a time of D as a byte alignment time. This can be calculated by the data rate, and after that, the time error is corrected with $T_2 - T_1 - D$ in the receiver node.

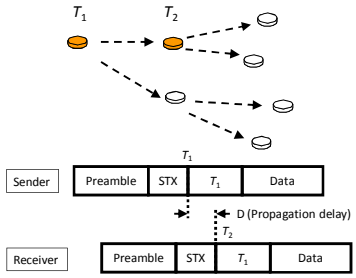

Figure 7. Flooding Time Synchronization Protocol.

When the synchronous process of the FTSP is completed, the time lag among sensor nodes is from 1 μs to 10 μs, and this value does not depend on the propagation delay [12]. However, [13] pointed out that the maximum time lag becomes 240 μs when executing 50 times synchronization per day using a 32.768 kHz real-time clock (TG-3530SA: EPSON TOYOCOM [14]). The time lag increases with a time after the synchronization because the oscillation frequency is affected by temperature and differs from each other. Unfortunately, the distributed delay-and-sum algorithm requires less than 100-μs accuracy [7]. To suppress the time fluctuation, we introduce a liner interpolation approach (Fig. 8). When the FTSP is carried out, every node stores the difference of timer value between a sender and itself. Using the stored data, the node can calibrate its counter value. In this way, the total time lag among nodes is reduces to 100 μs at maximum.

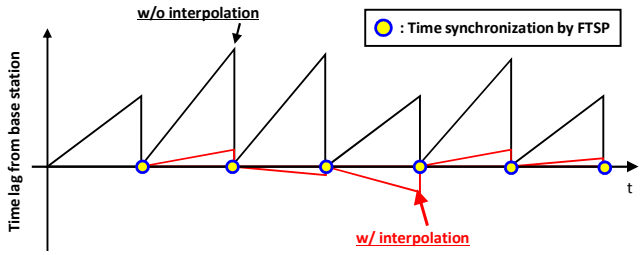

Figure 8. Linear interpolation for clock drift.

F. *Routing and Clustering*

In this subsection, we describe a routing protocol for the proposed sound acquisition (Fig. 9). The base station performs FTSP and NNT (Nearest Neighbor Tree) Protocol [15] simultaneously to create a spanning tree in the network. Then, every node makes transition to a sleep mode, in which only VAD and receiver circuits are activated.

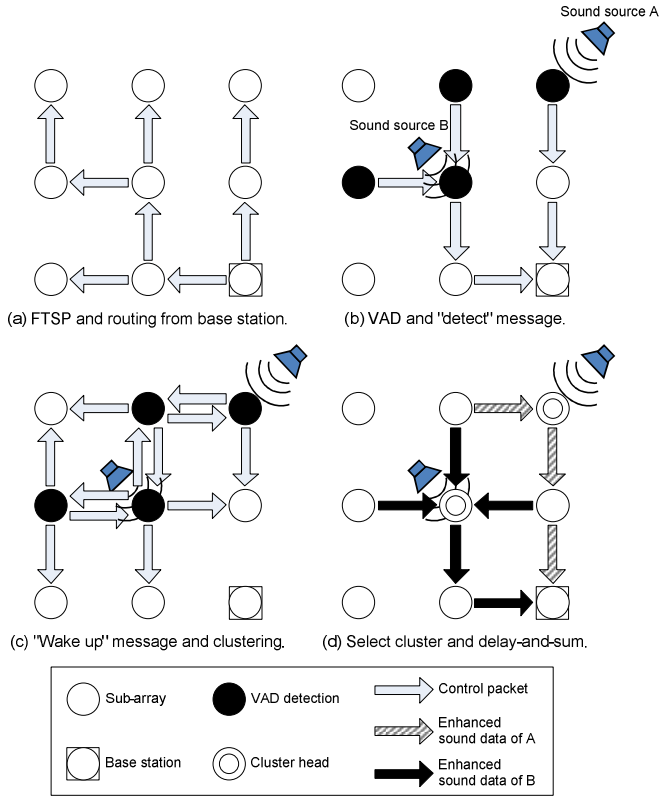

Figure 9. Routing and clustering example for multiple sound source acquisition (one-hop cluster).

After the VAD circuit in a certain node detects speech, the node prepares a “detect” message and a “wake up” message. The “detect” message is transferred to the base station according to the spanning tree, which is constructed in a setup phase. The “wake up” message is a broadcast message, which has a limited time to live and it creates a cluster around the node that detected the speech (this node possibly acts as a cluster head). Next, the nodes that received the “wake up” message and cluster heads perform the first-step sound source localization: They send a result of relative direction estimation to the base station.

As the second-step of the sound source localization, the base station chooses a cluster head which is the nearest of the sound source. If multiple sound sources are detected, multiple cluster heads will be selected. The base station broadcasts the estimated absolute locations to the selected cluster heads.

The nodes, which construct a selected cluster, carry out the local and global sound source separation with the delay-and-sum algorithm. Finally, the enhanced sound data are transferred to the base station via cluster head.

III. FPGA IMPLEMENTATION

To evaluate the proposed microphone array network, we made a prototype by using an FPGA (Field Programmable Gate Array) board. The prototype has the functions of the VAD, sound source localization, sound source separation, and wired data communication module.

A. Implementation

Fig. 10 shows the prototype, which consists of a SUZAKU FPGA board [16], 16-ch microphones, microphone amplifier board, A/D converter board, LED indicator for VAD, and two RJ-45 ports. The SUZAKU FPGA board includes a hardware PowerPC 405 processor. The size of the prototype is 30 cm × 30 cm. The 16-ch microphones are arranged in a 7.5-cm spaced grid array. The sampling rate is set to 16 kHz because the target of this system is the human voice that has a frequency range from 30 Hz to 8 kHz.

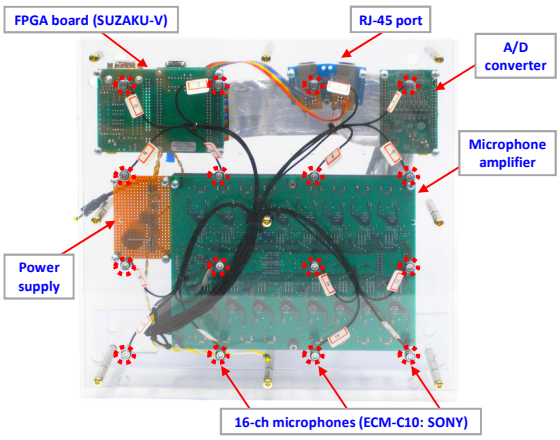

Figure 10. Prototype (one node).

The sub-arrays are connected by using a UTP cable. The 10BASE-T Ethernet protocol is used as a physical layer. In a data-link layer, we adopt LPL (Low Power Listening) protocol [17] to reduce power consumption.

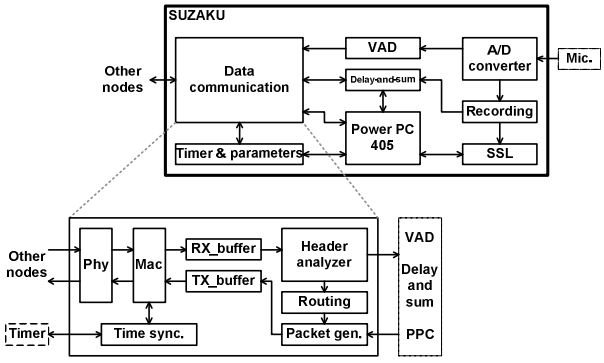

Figure 11. Block diagram of prototype.

Fig. 11 shows the block diagram of the prototype. The VAD module, FFT module for sound source localization, and delay-and-sum module are implemented as dedicated hardware in the FPGA. The other portion of sound source localization is processed by PowerPC. The physical layer and data-link layer for data communication is also implemented in the FPGA. To improve the precision of time synchronization, the FTSP and liner interpolation module must be implemented in the FPGA. The other network protocols and upper layers are handled by PowerPC.

B. Performance Evaluation

To verify the performance of the proposed system, we conducted experiments with three prototype sub-arrays. The sub-arrays are deployed as shown in Fig. 12. One sub-array, which is located in the center, is connected to a server PC as a base station. The network topology is two-hop linear topology to evaluate the multi-hop environment.

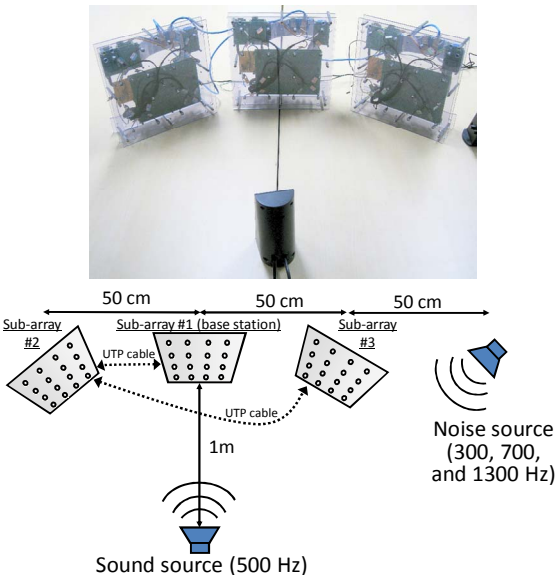

Figure 12. Experimental setup.

Fig. 13 shows a measurement result of two-hop time synchronization using the FTSP. The maximum time lag among sub-arrays was 1 µs just after the FTSP synchronizing process was completed. The maximum time lags between sub-arrays with and without the liner interpolation are 10 µs per minute and 900 µs per minutes, respectively.

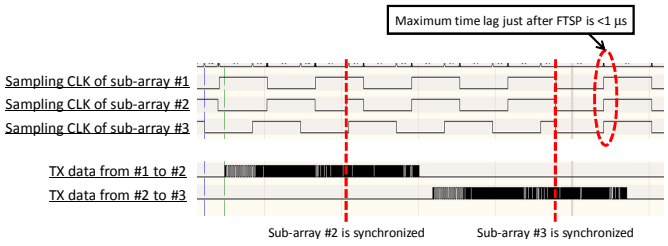

Figure 13. Measured wave form of time synchronization.

Next, we evaluated the sound acquisition using the distributed delay-and-sum algorithm. A 500 Hz sine-wave signal source and noise sources (300-Hz, 700-HZ, and 1300-Hz sine waves) were used. Fig. 14 shows the experimental results of the sound acquisition with one sub-array (1 ch and 16 ch) and three sub-arrays (48 ch). The signal is enhanced and the noises are decreased; that is, the SNR is improved as the number of microphones is increase. On the 48-ch condition, the noises of 300 Hz and 1300 Hz are dramatically suppressed by

20 dB without signal source (500 Hz) degradation. On the other hand, note that the noise of 700 Hz is slightly suppressed. This is because the interference caused by positions of the signal and noise sources. Fig. 15 explains the reason with simulation; the 700-Hz noise source can be hardly suppressed around the noise source position even in the 48-ch case. This problem can avoid by increasing the number of nodes.

We also confirmed that the sound acquisition using three sub-arrays can operate in real time.

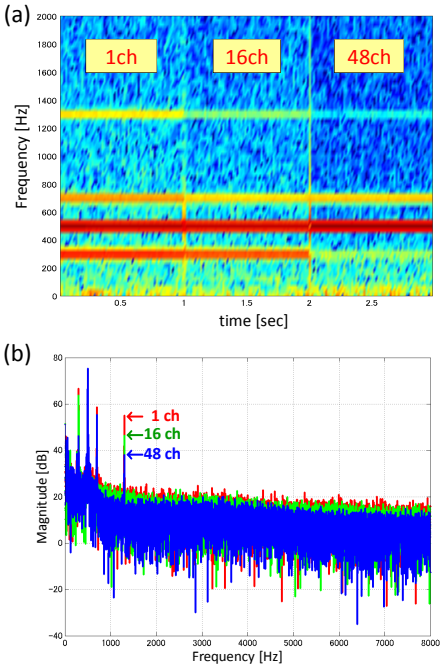

Figure 14. Experimental results of sound source acquisition: (a) time domain and (b) frequency domain.

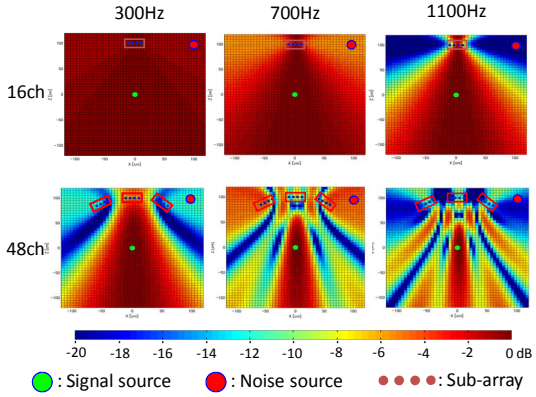

Figure 15. Simulation results of sound source acquisition.

IV. Conclusion

We proposed the distributed sound acquisition scheme for multiple sound sources. The microphone array network using 16-microphone sub-arrays was implemented as a prototype, and we verified the following operations with it on a node and network levels: 1) low-power VAD to activate the entire node, 2) sound-source localization, 3) time synchronization between the sub-arrays 4) autonomous distributed routing and clustering and 5) distributed sound-source enhancement. The experimental result of the sound-source enhancement shows an SNR improvement of 20 dB using 48 microphones.

Acknowledgment

This research was supported by the Semiconductor Technology Academic Research Center (STARC) and the Grant-in-Aid for JSPS Fellows, No. 21000333, the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan.

References

[1] M. Brandstein and D. Ward, “Microphone Arrays: Signal Processing Techniques and Applications,” Springer, 2001.

[2] Y. Tamai, S. Kagami, H. Mizoguchi, K. Sakaya, K. Nagashima, and T. Takano, “Circular Microphone Array for Meeting System,” Proc. IEEE Sensors, Vol. 2, pp. 1100-1105, 2003.

[3] T. Wakabayashi, K. Takahashi, H. Iwakura, “Independent Component Analysis using Large Microphone Array,” Technical report of IEICE. EA, Vol. 102, No. 322, pp. 29-34, 2002.

[4] H. F. Silverman, W. R. Patterson III, and J. L. Flanagan, “The Huge Microphone Array,” Journal of IEEE Concurrency, Vol. 6, No. 4, pp. 36-46, Oct-Dec. 1998 and Vol. 7, No. 1, pp. 32-47, 1999.

[5] E. Weinstein, K. Steele, A. Agarwal, and J. Glass, “Loud: A 1020-Node Modular Microphone Array and Beamformer for Intelligent Computing Spaces,” MIT, MIT/LCS Technical Memo, MIT-LCS-TM-642, 2004.

[6] H. Noguchi, T. Takagi, M. Yoshimoto, and H. Kawaguchi, “An Ultra-Low-Power VAD Hardware Implementation for Intelligent Ubiquitous Sensor Networks,” Proc. IEEE Workshop on SiPS, pp. 214-219, 2009.

[7] T. Takagi, H. Noguchi, K. Kugata, M. Yoshimoto, and H. Kawaguchi, “Microphone Array Network for Ubiquitous Sound Acquisition,” Proc. IEEE ICASSP, pp. 1474-1477, 2010.

[8] K. Kugata, T. Takagi, H. Noguchi, M. Yoshimoto, and H. Kawaguchi, “Live Demonstration: Intelligent Ubiquitous Sensor Network for Sound Acquisition,” Proc. IEEE ISCAS, pp. 1413-1417, 2010.

[9] H. Noguchi, T. Takagi, K. Kugata, M. Yoshimoto, and H. Kawaguchi, “Low-Traffic and Low-Power Data-Intensive Sound Acquisition with Perfect Aggregation Specialized for Microphone Array Networks,” Proc. IARIA SENSORCOMM, pp. 157-162, 2010.

[10] J. Benesty, M.M. Sondhi, and Y. Huang, “Handbook of Speech Processing,” Springer, 2007.

[11] F. Asano, H. Asoh, and T. Matsui, “Sound Source Localization and Signal Separation for Office Robot (Jijo-2),” Proc. IEEE MFI, pp. 243-248, 1999.

[12] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, “The Flooding Time Synchronization Protocol,” Proc. 2nd ACM SenSys, pp. 39-49, 2004.

[13] T. Takeuchi, Y. Otake, M. Ichien, A. Gion, H. Kawaguchi, C. Ohta and M. Yoshimoto, “Cross-Layer Design for Low-Power Wireless Sensor Node Using Wave Clock,” IEICE Trans. on Communications, Vol. E91-B, No. 11, pp. 3480-3488, Nov. 2008.

[14] “http://www.epsontoyocom.co.jp,” Epson Toyocom Corporation, (accessed 2011-02-25).

[15] Maleq Khan, and V.S. Anil Kumar, “Distributed Algorithms for Constructing Approximate Minimum Spanning Trees in Wireless Networks,” IEEE Trans. on Parallel and Distributed Systems, Vol. 20, No 1, pp. 124 - 139, Jan. 2009.

[16] “http://suzaku.atmark-techno.com/,” Atmark Techno, Inc., (accessed 2011-02-25).

[17] W. Ye, J. Heidemann, and D. Estrin, “Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks,” IEEE/ACM Trans. on networking, Vol. 12, No. 3, pp. 493–506, 2004.