Visual Sound Detector for the Hearing Impaired (May 2013)

Kunwoo Kim, Advanced Electronics Design Laboratory (520.738), Johns Hopkins University

Abstract— **This device aids deaf individuals who have difficulties in recognizing auditory alarms such as fire alarms, car honks, doorbells, and telephone calls. This portable device detects sound and re-alarms the user with a visual signal. LEDs on the board light up toward the direction of the sound. The color of the lighted LEDs represents the intensity of the sound.**

***Index Terms*—Sound Detection - Sound Localization - Visual Alarms**

I. Introduction

We encounter many auditory alarms today: phone rings, bells, car honks, smoke alarms, and speech calls. Deaf individuals in different degrees are impaired upon these signals because sound does not leave a visual trace. They seldom use tactile senses to sense these such as vibrations, but these are quite inefficient.

In this paper, I present a device that detects sound sources and visually show the direction and intensity of sound. Mainly, I was inspired by sound localization analysis by Professor Andreas Andreou of the Johns Hopkins University [1][2][3]. He used the cross-correlation technique to estimate the angle of the sound source.

However, I want to do a similar job, but under Arduino microprocessor's control. Arduino's memory is not capable of performing cross-correlation, since it requires a high sampling rate. Therefore, rather than using the temporal difference approach, I take the level difference approach, where I analyze the loudness difference.

One of the key advantages of the alternative approach is that it is more efficient (cost and energy). It does lack precision, but this device does not require a precise angle, since the microphones are 60 degrees apart. Also, it can localize multiple sound sources.

Translation of auditory signals to visual signals requires sound recognition, methods of visual display, knowing what are important sounds for deaf, and some functional issues to visualize non-speech signals [5][6][8].

The goal of this project is to create a cost-efficient, fast-processing, user friendly device that serves as a prototype that can prove its potential as a visual hearing aid.

This paper is conducted in Advanced Electronic Design Laboratory of the Electrical and Computer Engineering of the Johns Hopkins University. Kunwoo Kim is the only author of this article.

II. System Description

As represented in Fig. 1, the initial vision the system looks like a short cylinder. LEDs that show the angle and intensity of sound are placed on top, and the electronics and power source are placed beneath the LEDs. There are six electret microphones separated in 60 degrees of angle.

As the microphone detects a sound signal, it goes through the pre-amplifier circuit. Then, the Arduino undergoes the level difference process. Finally, the signal is sent to a comparator circuit that sorts out which LEDs to light up. This process is represented by Fig. 2.

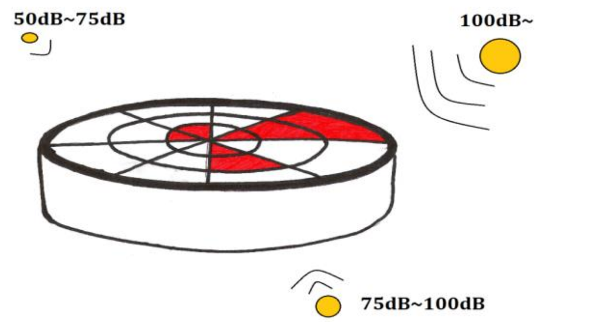

Fig. 1. System Description - Vision

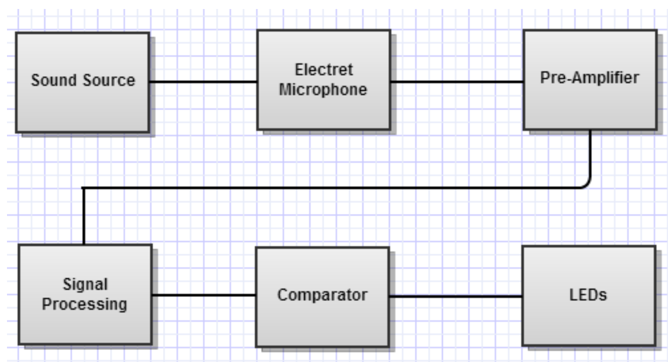

Fig 2. Simple Flowchart of the Basic Process

The LED array has two functions: portray direction and intensity. LEDs corresponding to each microphone has different colors to show different intensities. Then there are six of these arrays in line to show the direction. The combination of these two functions visualize the sound signal.

III. SETUP

A. Pre-Amplifier

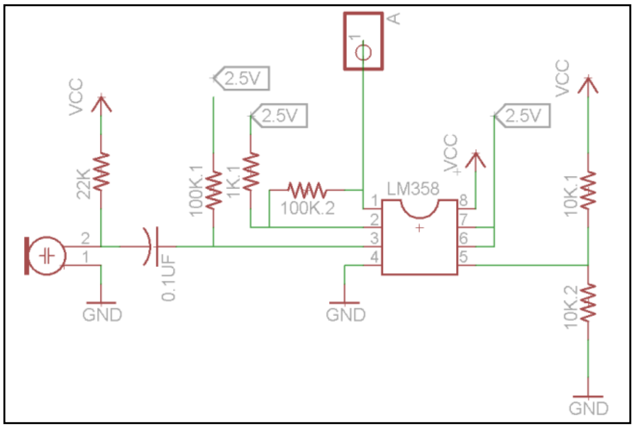

Fig. 3. Schematic of Pre-Amplifier.

This is a non-inverting amplifier with a 1:100 gain ratio. I use LM358 dual op-amp integrated circuit. Both microphone and the IC are supplied by 5V from the Arduino. The two 10K-ohm resistors are used as a voltage divider to supply 2.5V DC-offset. The output goes to Arduino's analog input.

B. Comparator

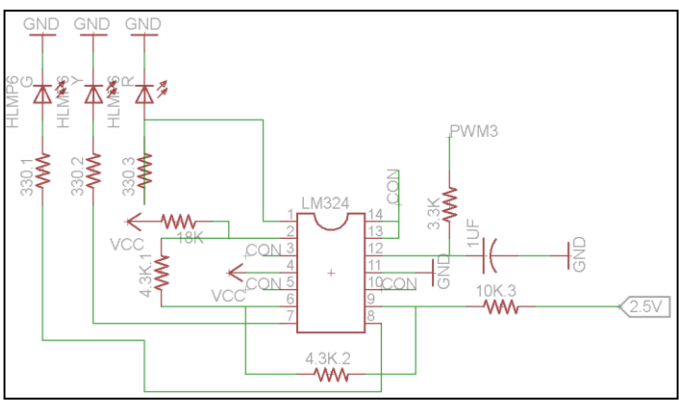

Fig. 4 Schematic of the comparator circuit.

I use LM324 quad op-amp integrated circuit. One op-amp is used as a bias to have the comparator circuit function as its own. There is a low-pass filter attached to the bias to convert PWM (Pulse Width Modulation) signal from the Arduino to voltage.

The rest three are then used for comparisons. The four resistor that go from VCC and 2.5V serve as a voltage divider. Therefore, with those resistor values, the first op-amp outputs voltage when the signal is about 45% of the maximum signal. The second op-amp outputs voltage when the signal is about 70% of the maximum signal. The third op-amp outputs voltage when the signal is about 90% of the maximum signal.

These circuits were simulated on LTSpice and on the breadboard. The resistors used for the voltage divider were precisely selected at this stage.

Once all the simulations were finished, these two circuit boards were put together and fabricated using the EAGLE software. The PCB board schematic is shown in Figure 5.

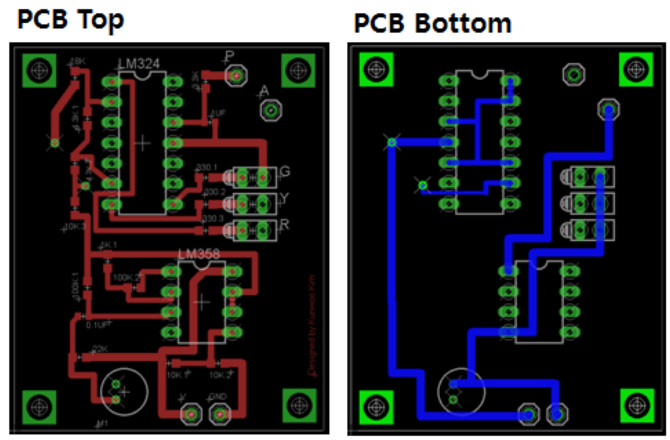

Fig. 5. Top View and Bottom View of PCB Boards

C. Digital Signal Processing

There are two options in digital signal processing: One source localization and Multi source localization. User of the device can choose either option with a switch.

One Source Localization

This option localizes the most significant sound source around the device. It allows the device to focus on the most critical source around it.

It takes 1 second to gather data while simultaneously visualizing the previous second. It reads the analog input signal from the pre-amplifier output. After testing on various time windows, I found out that those less than one second are prone to errors because of lack of data (since variance at each point is not small), and those more than one second may be prone to a signal that has changed its position within those time frames. If the signal value is below the DC offset, 2.5V, then the program subtracts it from the maximum voltage (5.0V) to preserve everything between 2.5V to 5.0V. This must be done since the LEDs turn on and off in reaction to the oscillating sound signal. Then it converts voltage value (5V max) to Pulse Width Modulation (255 Max). Lastly, all the values within the second are summed up while searching for the maximum value.

Next part is outputting the right maximum value to the right LEDs. After the second, it calculates which signal has the maximum value in the summation. I am using summation instead of calculating the peak-to-peak value because since all the signals are sampled (at rather a low rate), peak-to-peak value can be obscured. The maximum value is not always going to be sampled. Thus, while neglecting low voltage noise signals (defined as signals that do not affect LED), it sums up the noteworthy signals. Tests show that if I compare the microphone levels by their maximum values, the accuracy harshly falls because some signals as loud as a clap can make all six microphones to reach their maximum points. Also, additional tests show that if I compare the microphone levels by their integrated values, the accuracy is quite precise.

Then after choosing which microphone had the highest integrated value, it sends the stored maximum PWM value to the comparator. Thus one is to visualize the most significant signal at its maximum volume. Once signal is sent, all the variables reset to compute another second.

Multi Source Localization

This option localizes multiple signals at once. It allows the device to focus on various sound sources around it.

The procedure is similar to one source localization, but one difference is that it compares in groups of three adjacent microphones. If the middle signal is larger than the two adjacent ones, then that should be the direction where the signal is present. Thus having compared three at each time, the maximum number of signals it can detect is three. PWM signals are then sent to however many LED arrays they correspond to.

D. LED array

An LED array consists of 3 LEDs, green, blue, and red. Green lights up when the signal is over 3.1V, value based on the voltage divider. Yellow lights up when the signal is over 3.4V. Red lights up when the signal is over 3.75V. After the low pass filter on the comparator, the maximum value the PWM can give was 3.8V.

E. Sound Shadow Walls

Walls are put up around the PCBs and Arduino to put sound shadows on each microphone. Without these, the microphones were quite close to each other that the level difference was difficult to be accurate. Once a signal is sent, almost all the mics pick up the same values, and LEDs started to light up randomly. Thus like how human head attenuates sound signals from the left going to right, or vice versa, sound barriers wer put up to amplify the difference. The performance exponentially improved with them.

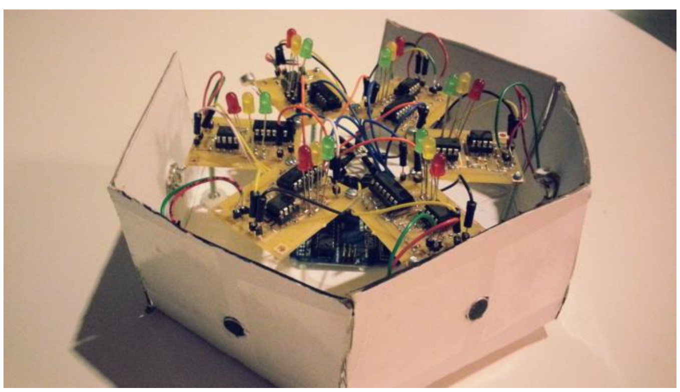

Fig 6. Compeleted Setup

IV. Discussion

Test results for one source localization in various environments are shown in Table 1.

	Quiet	Hallway	Lab
Success/Trials	44/50	33/50	35/50
Percentage	88%	66%	70%

Table 1. Test Results.

I had my cell phone alarm ringing in random positions. In quiet environments where almost all noise are under the threshold value for integration, there was a 88% success rate. Some microphones performed better than the other due to its position and proximity to the noise source. In the hall way, where there is a lot of reverberation, the performance was 66% , value that is lower than the quiet environment. Microphones closer to the wall seemed to perform better. In the lab environment where colleague members were around and lab equipments processing, I had 70% of success rate. Failures heavily depended on sudden noise generations.

Test results for multi source localization in various environments are shown in Table 2.

	Quiet	Hallway	Lab
Success/Trials	34/50	24/50	22/50
Percentage	68%	48%	44%

I had a computer generating a tornado warning siren, and my cell phone alarm both in random positions. The success rate of detecting one of the two sources was almost the same as that of the one source localization.

Here, lab environments had the worst performance, since there are many unexpected noise generated. This shows that multi source localization is prone to random noise more than reverberations in the hallway. Still, the success rate in the hallway is very low. Multi source localization is heavily affected by noise.

After the setup and going through various tests, I found out that ambient noise is a huge factor to errors. White noise, air-conditioning, heating, urban noise, and people chattering sometimes obscured the result because their noise level can be larger than the noise-threshold I set. If there is a dominant sound in the surroundings, localization seems to visualize the results well. However, if the signal is closer to the heightened noise level, sometimes localization directs to that point because I am relying on summation for localization.

Amplifying ratio of 1:100 may be too big. The microphone reaches maximum voltage more easily than expected. Thus choosing a different resistor value for amplification may improve the result.

The resistors for the voltage divider in the comparator circuit may need to be adjusted. The threshold value for the red LED is higher than the simulation. Also, the low pass filter can be adjusted after testing adjustments on the prototype.

The sound shadow wall is extremely effective. For localization that depends on level difference, some kind of mechanical barrier is crucial. I have the top and one side open in the figure, but creating a roof, the remaining wall, and dampening legs to reduce resonances within the enclosure, the performance is expected to be better.

More microphones mean higher resolution for the angle and that more sources can be detected on the multi source localization. Sensitivity of each mic should always be tested. For this design, it had a limit of six, since Arduino Uno has maximum six analog inputs and PWM outputs.

V. Conclusion

This device proves how level difference may be used for sound localization that can aid deaf individuals by translating auditory signals into visual ones. Though ambient noise and internal noise (microphones, operational amplifiers, etc) may obscure some of the wanted signals, the device does show degree of acceptable precision.

While most of the localization technologies ended up in papers and journals, this device serves as a prototype. As a cost-efficient, fast processing, low powered, easy interface device, this shows potential for a new kind of visual hearing aid.

REFERENCES

[1] G.C. Carter, "Time delay estimation for passive sonar signal processing," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-29, pp. 463-470, June 1981.
[2] P. Julian, A.G. Andreou, L. Riddle, S. Shmma, G. Cauwenberghs, "A Comparison of Algorithms for sound localization, 2003.
[3] Hasan Khaddour, "A Comparison of Algorithms of Sound Source Localization Based on Time Delay Estimation", 2011.
[4] T. Matthews, J. Fong, W. Ho-Ching, J. Mankoff , "Visualizing Non-speech Sounds for the Deaf", 2007.
[5] "Auditory and Visual Therapy Software", Sonido.Inc.
[6] W. Ho-Ching, J. Mankoff, J. A. Landay, "Can you see what I hear? The Design and Evaluation of a Peripheral Sound Display for the Deaf", 2003.