

2014 AASRI Conference on Sports Engineering and Computer Science (SECS 2014)

Android Handset VoIP Echo Cancellation

Ming-Che Yeh*, Cheng-Yu Yeh, Shaw-Hwa Hwang

NTUT, Electrical Engineering, #1, Sec. 3, Chung-hsiao East Rd., Taipei, Taiwan, ROC

NCUT, Electrical Engineering, #57, Sec. 2, Zhongshan Rd., Taichung, Taiwan, ROC

NTUT, Electrical Engineering, #1, Sec. 3, Chung-hsiao East Rd., Taipei, Taiwan, ROC

Abstract

In this study, the acoustic echo problem and cancellation method of Android-platform-based VoIP was analyzed and researched. The echo problem is consequential for Android phone-based VoIP applications. However, in the traditional 2G/3G circuit-switch (CS) mode, the acoustic echo is cancelled by the hardware chip, and the quality of communication is adequate. In the future 4G/LTE environment, all-IP packet communication will be achieved. The IMS/VoIP will become the main application for personal speech communication systems. Before the introduction of 4G with echo cancellation built into its chip, the Android-based VoIP on 3.5G/WiFi will continue to undergo the aforementioned acoustic echo problem. This paper presents an analysis of the acoustic echo on Android-based VoIP and proposes a simple echo cancellation method.

© 2014 The Authors. Published by Elsevier B. V. This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/3.0/>).

Peer-review under responsibility of Scientific Committee of American Applied Science Research Institute

Keywords: android; echo cancellation; echo suppressor;

1. Introduction

On the Android platform, the microphone and speaker environment creates the acoustics. Two acoustic echo paths can be determined. The first path is the air, and the second is the handset case. In the traditional

Ming-Che Yeh. Tel.: +886-919-097744;

E-mail address: tp26610@gmail.com

2G/3G circuit switch communication mode, the acoustic echo cancellation function is built into the GSM chip. The LMS algorithm is commonly used for echo cancellation. However, Android-based VoIP is a pure software app. The communication protocol, SIP/RTP, the speech codec, G.711/G729, and acoustic echo cancellation must all be implemented by software. Acoustic echo cancellation is difficult for two reasons: First, the LMS-based echo cancellation algorithm requires high computational power. Second, the record/play process on the Android platform is conducted frame by frame. The minimal frame size is 10 ms (80 sample points), thus lengthening the filter length and computation of the LMS algorithm. Previous echo cancellation methods that have been proposed for the Android platform are listed as follows:

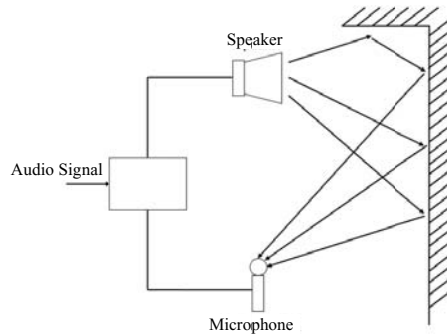


Fig.1. Acoustic Echo

1. Android API: Class acoustic echo canceler

This was added to the Android API level 19. In this study, attempts were made for this function to be implemented into the authors' VoIP application. However, it did not work, or the echo cancellation was not evident.

2. Android GSM codec

In this study, attempts were made to use a traditional 2G/3G circuit switch (CS) session. The echo was not present and the voice quality was excellent. Hence, whether the echo canceller was implemented in the hardware or the chip had to be determined by reading the android source code and retrieving the segment of code for the CS session. The segment of code was then used to simulate the echo function in the hardware. The echo was cancelled by the hardware and then passed to the VoIP.

3. Software computation: Least mean squares [1]

This algorithm is used to capture echo audio samples. Most android system audio drivers capture 10 ms of audio data at a time slot. The LMS requires a high-order buffer length and computational power, which seems impossible. The time delay of voice over VoIP is too long for communication. Real-time communication is the most crucial feature of the VoIP system.

4. Software computation: Echo suppressor [2]

This is the simplest approach to implement, and yields a high-quality voice when parameters are coordinated perfectly. The echo canceller determines the characteristics of the audio data relationship between the playing and recording of an audio device. The following section details this approach.

2. Echo suppressor algorithm

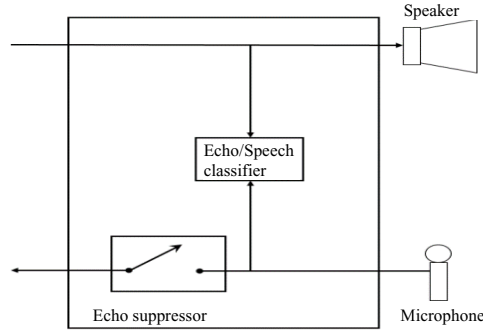


Fig.2. Block diagram of an echo suppression system

In the echo suppressor approach, each sample value is S , and the energy of each frame is E . Each frame contains 80 sample points.

$$F = \{S_1, S_2, S_3, \dots, S_{80}\}$$

The value F expresses an audio frame. The energy of each frame is

$$E = \sqrt{S_1^2 + S_2^2 + S_3^2 + \dots + S_{80}^2}$$

In comparing the energy of playback and recording, numerous situations and rules can be concluded. The playback frame energy is E_p and the recording frame energy is E_R . Audio frame energy is a sequence that depends on the frame (10 ms):

$$E_p = \{E_p(t), E_p(t-1), E_p(t-2), \dots, E_p(t-n)\}$$

$$E_R = \{E_R(t), E_R(t-1), E_R(t-2), \dots, E_R(t-n)\}$$

1. If $E_p(t) \leq E_R(t)$,

The audio frame energy does not echo. Even recording data contain an echo element, but are masked by the recorded speaking voice. Thus, the echo suppressor is disabled in this situation.

2. If $E_p(t) > E_R(t)$,

This indicates that a remote user is speaking. A person who is speaking they does not hear the voice of the listener. This is a real situation in human communication. One speaks while the other listens.

Based on this characteristic, the echo suppressor is enabled in this situation.

The previous discussion does not consider the effect of time delay. In Android audio devices, audio is not sampled in real time. The recording has an A/D converter and the playback has D/A converter. The time delay causes the echo suppressor to cancel the real voice, not the echo. Therefore, the time delay between recording and playback must be determined. The conditional equation is modified as follows:

1. If $E_p(t) \leq E_R(t-D)$, disabling the echo suppressor.

2. If $E_P(t) > E_R(t - D)$, enabling the echo suppressor.

D is the time delay between sound playback and recording. The impulse response of playback and recording is presented in Fig. 3.

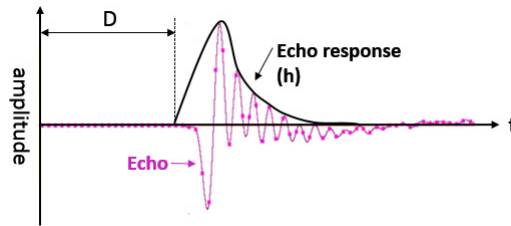


Fig.3. Echo response.

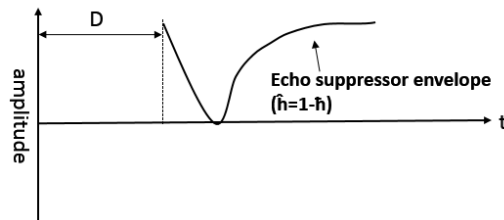


Fig.4. Echo suppressor envelope.

The echo response could be determined in this study, and an adverse echo response was generated accordingly. The involved equation can be modified as follows:

1. If $E_P(t) \leq rE_R(t - D)$, disabling the echo suppressor.
2. If $E_P(t) > rE_R(t - D)$, enabling the echo suppressor envelope function.

Where r is the rate of decay in echo, $\hat{h}[\cdot]$ is the echo suppressor envelope, D is the time delay, and $\mathbf{h}[\cdot]$ is the normalized response from $\mathbf{h}[\cdot]$.

3. Experiment results

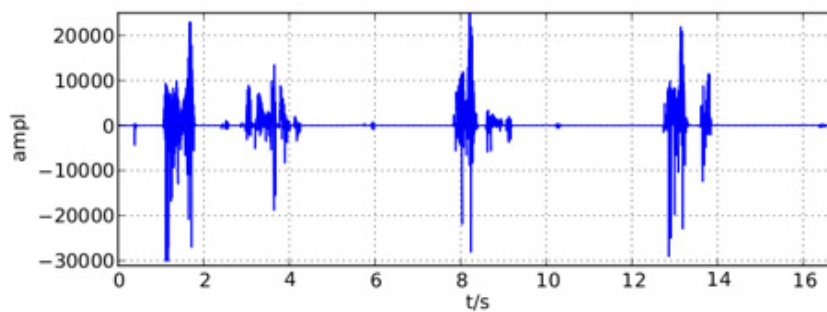


Fig.5. Voice in playback

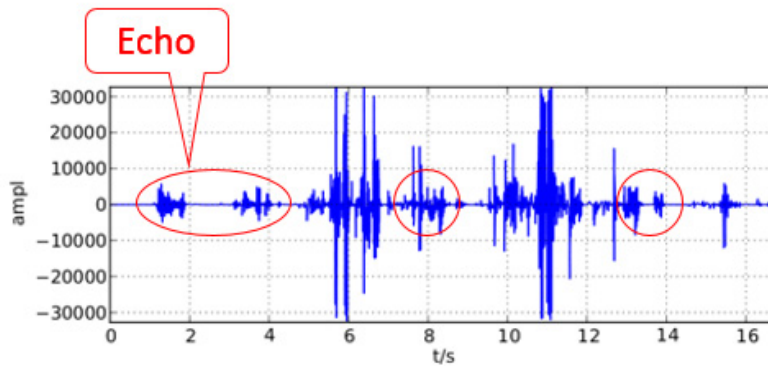


Fig.6. Voice in recording

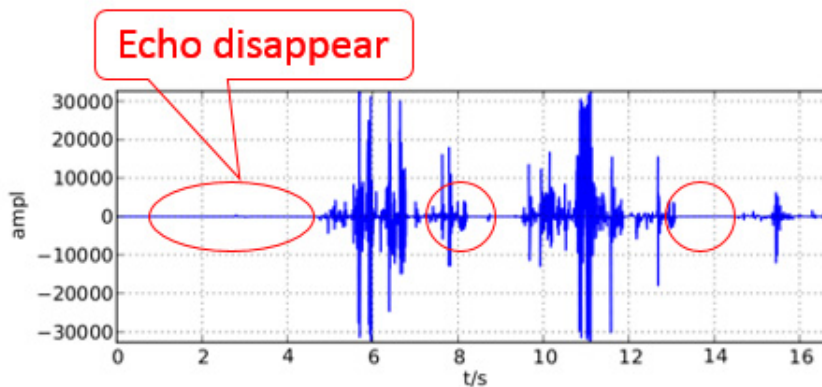


Fig.7. Voice recorded through the echo suppressor

Fig. 5 shows the original playback voice. Fig. 6 depicts the microphone recording, which contains some echoes. Evidently, as shown in Fig. 7, the playback echo was cancelled and disappeared. The experimental result shown in Fig. 7 confirms the high performance of the proposed approach.

4. Conclusion

This study analyzed the acoustic echo for Android-platform VoIP. A simple acoustic echo cancellation method is proposed. The experimental results confirm high performance.

References

- [1] J. Benesty, T.Gänsler, D.R. Morgan, M. M. Sondhi, and S.L. Gay 2001. Advances in Network and Acoustic Echo Cancellation
- [2] Eberhard Hänsler, and Gerhard Schmidt 2004. Acoustic echo and noise control: a practical approach.