Operational control of the multimedia player of smart phones using a Kinect voice-sensing scheme

*Ing-Jr Ding, Chong-Min Ruan and Jia-Yi Shi

Department of Electrical Engineering, National Formosa University, Taiwan

*email: eugen.ding@gmail.com

Keywords: Kinect Voice-Sensing; Voice Control; Layered Hierarchy Tree; Speech Recognition; Multimedia Player

Abstract. Voice control has been a popular human machine interface in electrical equipments including the smart phone device. Voice control on smart phone still suffers from inconvenient operations due to the limitation of the extremely short distance between the human operator and the phone device. This paper develops a Kinect voice-sensing scheme with a layered hierarchy tree for command operations of the multimedia player on the smart phone. Presented voice control by Kinect voice-sensing will provide a more convenient way for smart phone operations where the human operator can still control the smart phone using voice commands in the situation that the phone device is not carried by the user and far away from the user. Experiments on the multimedia player operations using a series of voice commands sensed by Kinect demonstrated the superiority of the presented Kinect voice-sensing scheme on smart phone control.

Introduction

Speech recognition has been a matured technique recently [1]. Command control is the most popular application among all possible speech recognition applications. Voice command control by speech recognition has been a standard function in the smart phone device. Application programs including the widely-used multimedia player could be operated by the voice command utterance of the human user. However, the merit of such the human computer interface will not be maintained in the situation that the smart phone device is a little far away from the user and is not carried by the user. To overcome this problem and to tolerate a longer distance recognition task between the human user and the target smart phone device, a voice-sensing speech recognition scheme with the deployment of voice sensors can be employed. Figure 1 depicts such the concept of voice-sensing command recognition for operating the multimedia player application program of smart phones.

In this study, the popular Kinect sensor is adopted for performing voice-sensing. The Kinect sensor is made by Microsoft [2, 3] and is well-known for its abilities in the applications of the person's gesture recognition [4-8] and speech recognition [9]. Most of Kinect-related studies focus on image-sensed gesture recognition and rare investigations aim at voice-sensed speech recognition. In fact, Kinect could be used to be an excellent voice receiver for performing speech recognition due to an incorporated microphone array sensor. This work employs the Kinect microphone array to sense the voice around the arranged environment, and voice command keywords made by the human user will be recognized. The recognized voice command could be then immediately transmitted to the smart phone device that is a little far away from the user for remotely operating the corresponding function of the multimedia player APP.

Voice Control by a Kinect Voice Sensing Scheme

The developed Kinect voice-sensing scheme for controlling smart phone applications by voice commands could be divided into three technical components, which are voice sensing and voice command recognition by Kinect, control command transmission by Bluetooth communication, and control command reception and function trigger on the target player. These three processing components are described as follows.

Fig.1. A scenario of Kinect voice-sensing speech recognition for operations of the smart phone multimedia player

The section describes voice sensing and voice command recognition by Kinect. The Kinect device developed by Microsoft has been well-known for its effectiveness on pattern recognition including speech recognition in this work. This study employs the Kinect sensor for sensing the person's uttered voices and the Kinect software development kit (Kinect SDK) for rapidly establishing the voice command recognition system [3, 10, 11].

For voice sensing, the Kinect sensor mainly contains a microphone array which is composed of four microphones. Four sets of voice data from these four microphone voice receivers will be combined into a single set of voice data by the Microsoft microphone array SDK technique [3]. The combined voice data acquired from the sensed environment will be continuously evaluated if the data involves the segment of voice control command keywords.

For voice control command recognition, the Kinect speech recognition SDK will be useful and convenient for the program developer. To establish a Kinect-SDK speech recognition system, the primary work of the developer is to edit the "XML file," which is a referenced table for recognition and consists of all voice command utterances. For example, in this work of multimedia player control, the table will contains "May I play the song," "Could you decrease the volume," "Stop the song," "Pause the song," "May I replay the song,"…, etc. Such the XML file design for arranging voice command utterances will be crucial and an improper XML editing will lead to the substandard performance on recognition processing time.

To have a high performance on system response time when using Kinect-SDK speech recognition, this paper proposes a hierarchical tree scheme which could be embedded into the devise of XML voice command editing. Figure 2 depicts a four-layered hierarchy tree is proposed for rapid spotting of voice command keywords. As could be seen in Fig. 2, the hierarchy tree includes four processing layers, the layer of auxiliary verb terms, the layer of subject terms, the layer of verb terms and the layer of object terms. In this work of multimedia player voice control, a complete voice control sentence made by the user could be "May I play the song," and two keyword terms are included in this command sentence, which are the verb 'play' and the object "the song." As shown in Fig. 2, each of auxiliary verb terms, subject terms, verb terms and object terms in a complete command utterance will be independently arranged into the same processing layer on XML designs, and therefore, the nodes with the same category property will be recognized separately. When keyword terms for target control are appeared and recognized, the label denoting the functional operation of the recognized keyword terms is searched and such the label will be transmitted to the target smart phone device for multimedia player control, which will be described in the following section.

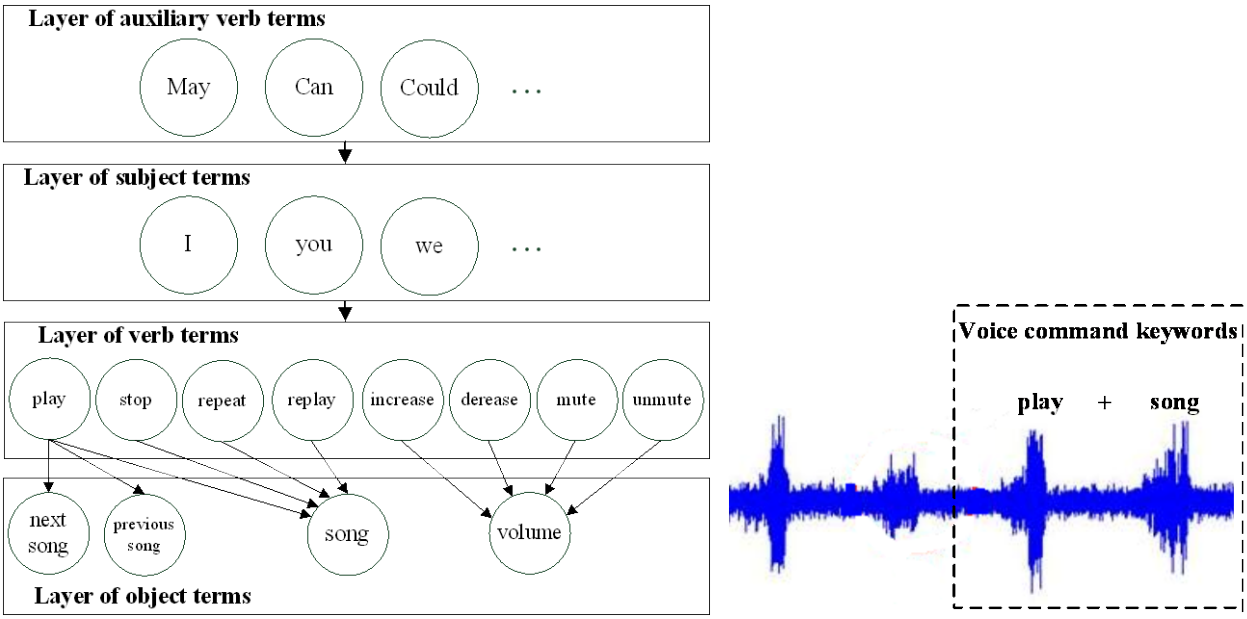


Fig.2. The developed Kinect voice-sensing voice control scheme employs a four-layered hierarchy tree for efficient keyword voice command-spotting on Kinect speech recognition where the proposed hierarchy tree and keywords spotting are in Fig. 2(a) and Fig 2(b).

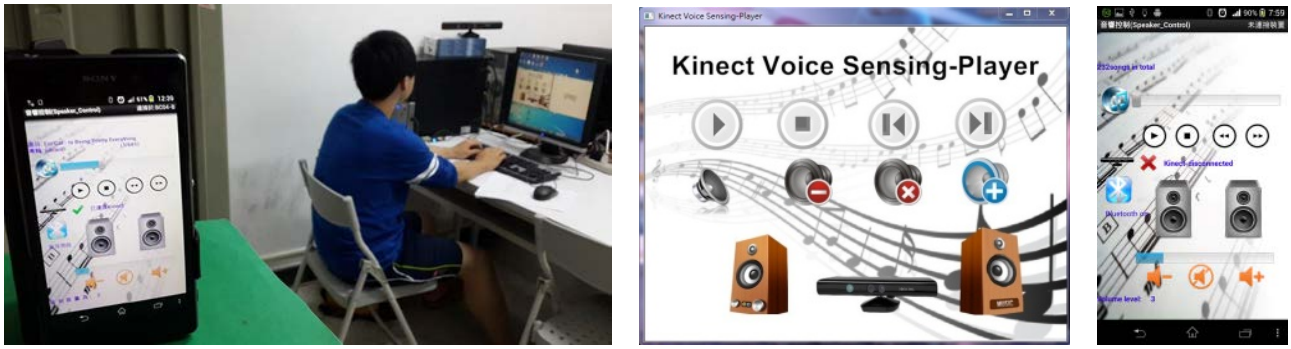
Control command transmission by Bluetooth communication in the presented voice control-based Kinect voice sensing scheme will be provided in this section. The recognized voice command keywords using Kinect-SDK speech recognition with the presented hierarchical classification tree scheme is then sent to the target smart phone device for the player operations. In this work, Bluetooth wireless transmission is extremely appropriate for establishing a communication line between the Windows platform with the Kinect sensor and the smart phone device and therefore adopted. A series of voice command keywords for the multimedia player operations on the smart phone device, “play the next song,” “play the previous song,” “stop the song,” “repeat the song,” “replay the song,” “increase the volume,” “decrease the volume,” “mute the volume,” and “unmute the volume,” are given proper labels for representing the corresponding player operations. For providing an effective and efficient command transmission on the Bluetooth communication line, the label for each of voice command keywords is designed as the single text symbol. For example, these voice command keywords, “increase the volume,” “decrease the volume,” “mute the volume,” and “unmute the volume,” are represented by the text symbols, ‘i,’ ‘d,’ ‘m,’ and ‘u,’ respectively. When one of these voice command keywords is appeared and recognized, the corresponding symbol of the voice command keywords is sent to the target phone device for voice command interpretations on the built Bluetooth communication pair.

The section provides control command reception and function trigger on multimedia player. This processing component in this section will perform control command receptions and then the functional operation trigger on the target multimedia player application program. In this study, the multimedia player APP is developed in the smart phone with the Android platform. For rapidly establishing the Bluetooth function on the target Android smart phone device, the released Android software development kits are employed in this work [10, 11]. The Bluetooth component for receiving the text symbol of each set of the voice command keywords is properly integrated into the target multimedia player APP. In this study, a text symbol table for representing all voice command keywords is embedded into the design of the multimedia player. When the smart phone device obtaining a text message from the Bluetooth-connected Windows Kinect platform, the received message is firstly translated to get the corresponding player operation by looking up the built text symbol table. Due to a limited number of voice command keywords designed in this Kinect voice-sensing system, a simple linear searching method is adopted for decoding the received text

symbol. The interpreted voice command operation from the text symbol table is then used to trigger the corresponding function of the multimedia player application program.

Experiments and Results

Experiments on presented Kinect voice-sensing for operating multimedia player of smart phones are divided into two phases, the system establishment phase and the established system testing phase. Two experimental phases are performed in the environment of laboratory offices. In the phase of system establishments, different to those speech recognition development tasks without the use of the Kinect speech recognition SDK, speech recognition for classifying the sets of voice command keywords is established rapidly where no training speakers are requested for voice collections and no speech databases are required. In the phase of system testing, presented Kinect voice-sensing with a hierarchical classification tree could efficiently extract the useful voice command keywords in the sensed laboratory environment and remotely control the player program on the smart phone device using this Kinect-acquired voice command keywords. Figure 3 shows the developed system in the laboratory environment. As could be seen in Fig. 3, the smart phone with the multimedia player is a little far away from the voice command maker in the Kinect voice-sensing environment and could still be able to be finely controlled by the given voice command.



(a) (b) (c)

Fig.3. Kinect voice-sensing for operating multimedia player on smart phone far away from the voice command maker where application scenario, Kinect sensing-interface on Windows platform and multimedia player interface on the end device are in Fig. 3(a), Fig. 3(b) and Fig. 3(c).

Conclusion

In this paper, a Kinect voice-sensing scheme with a hierarchical layered tree of auxiliary verb terms, subject terms, verb terms and object terms is developed for remotely controlling the multimedia player application program on the smart phone device. Compared to conventional speech recognition without the use of Kinect voice-sensing, the presented scheme will provide a better way for controlling the operation of the multimedia player. The main competitively merit of this proposed scheme is that the human operator can still effectively control the player on the smart phone using voice commands in the situation that the phone device is not carried by the user.

Acknowledgement

This research is partially supported by the Ministry of Science and Technology (MOST) in Taiwan under Grant MOST 103-2221-E-150-046.

References

[1] I.-J. Ding and Y.-M. Hsu, “An HMM-like dynamic time warping scheme for automatic speech recognition,” Mathematical Problems in Engineering, vol. 2014, Article ID 898729, 8 pages, 2014.

[2] Z. Zhang, “Microsoft kinect sensor and its effect,” IEEE Multimedia, vol. 19, no. 2, pp. 4–10, 2012.

[3] I. Tashev, “Kinect development kit: a toolkit for gesture- and speech based human-machine interaction,” IEEE Signal Processing Magazine, vol. 30, no. 5 , pp. 129–131, 2013.

[4] I.-J. Ding and C.-W. Chang, “An eigenspace-based method with a user adaptation scheme for human gesture recognition by using Kinect 3D data,” Applied Mathematical Modelling, DOI: 10.1016/j.apm.2014.12.054.

[5] K. Qian, J. Niu and H. Yang, “Developing a gesture based remote human-robot interaction system using Kinect,” International Journal of Smart Home, vol. 7, no. 4, pp. 203–208, 2013.

[6] K. Arai and R. A. Asmara, “3D skeleton model derived from Kinect depth sensor camera and its application to walking style quality evaluations,” International Journal of Advanced Research in Artificial Intelligence, vol. 2, no. 7, pp. 24–28, 2013.

[7] S. Bhattacharya, B. Czejdo and N. Perez, “Gesture classification with machine learning using Kinect sensor data,” Proc. International Conference on Emerging Applications of Information Technology, pp. 348–351, 2012.

[8] S. Celebi, A. S. Aydin, T. T. Temiz and T. Arici, “Gesture recognition using skeleton data with weighted dynamic time warping,” Proc. Computer Vision Theory and Applications, 2013.

[9] G. Galatas, G. Potamianos and F. Makedon, “Audio-visual speech recognition incorporating facial depth Information captured by the Kinect,” Proc. the European Signal Processing Conference (EUSIPCO), pp. 2714–2717, 2012.

[10] T. Norbye, “Android studio 1.1 preview 1 released,” Android Tools, Google**,** 2015.

[11] X. Ducrohet, T. Norbye and K. Chou, “Android studio: an IDE built for Android,” Android Developers Blog, Google**,** 2013.