Poster Abstract: Proximity-Triggered Speech Recognition in Mobile Cloud Computing

Chia-Yu Lin
Department of Computer Science
National Chiao-Tung University
Hsin-Chu, 300, Taiwan

Yu-Jia Chen
Department of Electrical and Computer Engineering
National Chiao-Tung University
Hsin-Chu, 300, Taiwan

Li-Chun Wang
Department of Electrical and Computer Engineering
National Chiao-Tung University
Hsin-Chu, 300, Taiwan

Yu-Chee Tseng
Department of Computer Science
National Chiao-Tung University
Hsin-Chu, 300, Taiwan

ABSTRACT

Speech recognition applications have become very popular and have changed our behaviors of using Smart phones. However, users still have to initiate the voice recording by touching the small icons on the screens of smart phones. This action brings users great inconvenience especially for those who are driving. To overcome this issue, we propose a proximity triggered “no-touch” mechanism for smart phones by using proximity sensors to initiate mobile applications without the need of touching the screen. In this paper, we present the system framework of the proposed “no-touch” speech recognition initiation mechanism for cloud-based speech recognition applications on Android platform.

Categories and Subject Descriptors: C.5.3 [COMPUTER SYSTEM IMPLEMENTATION]: Portable devices

General Terms: Design, Experimentation

Keywords: speech recognition, proximity sensor, cloud computing

1. INTRODUCTION

Smart phones become popular with the help of hardware improvement and new functions resulted from many sensors. One important application for smart phones is speech recognition [1], which allows drivers to be hand free. Recently, the trend for speech recognition in mobile phones is to involve with cloud computing [3]. Since speech recognition requires a lot of computing resources, cloud-based speech recognition becomes inevitable for resource-limited mobile phones. In cloud-based speech recognition, a server can perform pattern matching in parallel and stores massive sampling audio files on the cloud database. However, current speech recognition applications have two obvious drawbacks. First, users have to focus on the phone screens and touch the small icons to initiate the recording process, which leads to a great deal of operational inconv-

ience especially for those who are driving. Second, a large amount of power is consumed since the phone screens have to keep lighting for the recording buttons.

In this paper, we propose a proximity triggered “no-touch” mechanism for smart phones that not only enhances the flexibility and convenience for smart phone users but also reduces the power consumption [2]. Our main contribution is a system framework for cloud-based speech recognition applications that exploits the proximity sensor to trigger speech recognition. To our knowledge, this is the first paper to discuss the implementation flow of proximity sensors driven cloud-based speech recognition on Android platform.

2. SYSTEM DESIGN

In proximity sensors driven speech recognition system, we can use the proximity sensor value to differentiate waving hands and taking the handsets. Figure 1 shows an example for the proximity sensor value of these two actions. For waving hands, the sensor value versus time is a triangle-like shape. For taking phones, the sensor value is one for a while. Therefore, the system can detect whether users is taking handsets by the change of sensor value. Adopting this method can reduce the possibilities of mis-triggering sensors.

The proximity sensors driven speech recognition service involves a cloud server for mobile social network services, called JOIN [5], cloud-based speech recognition server platform and application interface in the client’s handset. In JOIN, users can invite friends to attend an activity and se-
select the restaurants and dating time by vote, as shown in Fig. 2(a).

Fig. 2(a) shows the system architecture of proximity sensor driven cloud-based speech recognition in a mobile social network service. This system consists of five main components: text to speech (TTS) engine, proximity sensors, recorder on handsets, mobile social network server, and cloud-based speech recognition server. TTS engine can read out the message received from the mobile social network server. If there is a vote, users can put their palms on the top of proximity sensors. The proximity range sensor [6] is used to start an initiation event by detecting the distance between the customer’s palm and the handset. If an initiation event is detected by proximity sensors, the recorder will be activated to store user’s audio data. As receiving audio files from the handsets, speech recognition servers use massive data in the cloud database to perform pattern matching, and return the matching results. Speech recognition requires a lot of resource to search and translation computing, so speech recognition servers are built on cloud platform. The correct recognition results checked by users are sent to the mobile social network server. The mobile social network server can broadcast the polling result and manage activities between user groups, Fig. 2(c). Because the mobile social network server has to store and access amount of users’ information and activities, it was implemented on Microsoft SQL Azure.

3. DEMONSTRATION AND EVALUATION

We use HTC Desire to be the mobile platform. The OS system of Desire is Android 2.3. Users can activate speech recognition by waving hands close to small-sized ear cap which is located proximity sensor, as shown in Fig. 2(d).

Proximity sensors can be triggered by users without focusing on the screen. We find that the proximity sensor consumes less power from Table I. Thus we select proximity sensors to trigger speech recognition. In addition, since a higher a sensing frequency may cost more power, there is a tradeoff between the power consumption and the sensing successful rate, i.e. the rare that proximity sensors can successfully detect the waving action. Thus, finding a suitable sensing frequency is an important issue in mobile application. We set the sensing frequency which are FASTEST (sensing interval is 0 ms), NORMAL (sensing interval is 200 ms), and without using sensors but lighting screen to compare the power consumption. Table II shows the available time for 1% power consumption of three different cases. From Table II, we can find out NORMAL mode brings acceptable sensing successful rate and reduce the power consumption, compared with the case of lighting screen. Thus, we achieve power saving and convenient at the same time in this implementation.

4. REFERENCES