A Proximity Sensor Based No-Touch Mechanism for Mobile Applications on Smart Phones

Chia-Yu Lin, Yu-Jia Chen, Li-Chun Wang, and Yu-Chee Tseng
Department of Electrical and Computer Engineering
National Chiao Tung University, Taiwan

***Abstract*— Smart phones with touch screens have become very popular and have changed our behaviors of using handsets. However, using touch screens is not safe for mobile phone users especially when they are driving cars. Thus, many applications using smart phones cannot be initiated because users who are driving the cars cannot easily touch the small icons on the screens of smart phones. To overcome this issue, we propose a proximity sensors based "no-touch" mechanism for smart phones by applying proximity sensors to initiate mobile applications without the need of touching the screen. We will discuss how to implement the proximity-sensors driven "no-touch" mechanism in Android platform and investigate its performance issues regarding detection accuracy and power consumption. Speech-oriented applications using the proposed "no-touch" mechanism on Android is also demonstrated in this paper.**

***Index Terms*— proximity sensor, speech recognition, cloud computing**

I. Introduction

Nowadays people almost carry mobile phones anytime anywhere. Because of the improvements in hardware, mobile phones with many sensors can detect human actions. One of key breakthroughs in smart phones is the touch screen, on which users can initiate the applications by just touching the small application icons. However, it is inconvenient for mobile users to touch the small application icons when they are driving cars. Thus, how to develop a no-touch mechanism to start applications become an interesting issue. Some of speech oriented applications in smart phones, such as Siri in iPhone, also drive the technology trends of smart phons in this direction. Nevertheless, a flexible "no-touch" application initialization scheme for smart phones has not seen too often in the literature.

In this paper, an application initialization mechanism without the need of touching the screen of smart phone is proposed by using proximity sensors. With a pair of infrared (IR) transmitter and receiver positioned at the top of a handset, proximity sensors can compute the distance between the object and the sensor according to the reflection of IR light. Current applications of proximity sensors in smart phone are mainly focused on power saving. As long as the distance between objects is very close, proximity sensors can help detect this event and turn off the light of touch screen. In this paper, we find that proximity sensors can have a new application of initiating applications without touching the screens. We call this proximity-based application initialization mechanism as the "no-touch" mechanism in this paper.

The proposed "no-touch" mechanism consists of two basic functions: (1) event differentiation and (2) speech confirmation. We propose the "Waving hands testing mechanism (WTM)" to distinguish the event initiation from taking handsets. Besides, we use speech confirmation mechanism to check the recognition result after recognizing for enhancing the speech recognition accuracy. The combination of WTM and speech confirmation can ensure "no-touch" mechanism executing correctly and smoothly. In addition, we implement the proximity-based "no-touch" mechanism for speech oriented applications in Android phones. We will also show the experimental results of smart phones using "no-touch" mechanism in speech oriented applications to demonstrate the improvements of recognition accuracy compared to the Google speech recognition function.

The remaining parts of this paper are organized as follows. Section II introduces the related background on proximity sensors and speech recognition. Section III discusses the system architecture of this designed work. In Section IV, we discuss the implementation features and issues of the proximity-based "no-touch" mechanism for speech oriented applications. In Section V, we show experimental performance results of the WTM and speech confirmation mechanism and find out the most suitable sensing frequency for switching on proximity sensors. In Section VI, we give the concluding remarks and suggest the potential future research directions in the cloud-based speech recognition technology.

II. BACKGROUNDS

To make a touch-screen smart phone become "no-touch" and maintain its original functions, a mobile phone needs to combine many sensors, such as proximity sensors, acceleration sensors, gravity sensors and light sensors. In [1], proximity sensors were first proposed to control the voice operations in handsets. Although the work in [1] was not implemented and many performance issues are remained unsolved, it did inspire us to take advantage of proximity sensors to design a "no-touch" mechanism to issue control commands to the applications of smart phones.

The principles of proximity sensors are introduced as follows. In [4], sensing techniques for mobile interaction were

978-1-4673-1881-5/12/$31.00 ©2012 IEEE

studied, including proximity range sensors, tilt sensors, and touch sensors, etc. Proximity range sensors are placed near the mini-sized ear cap of handset, which include a pair of infrared transmitter and receiver. After the transmitter of the proximity sensors emit IR light, the light is reflected by an object near proximity sensors. The time duration of transmitting IR light and receiving its reflected light can be transformed to the distance between the object and the proximity sensor. If the sensing distance of the proximity sensor is changed, an event can be triggered. The sensing distance of proximity range sensor is from 5 cm to 25 cm. Right now, proximity sensors in mobile phones are enabled as soon as the function of voice conversations are activated. If the proximity range sensor finds that the mini-sized ear cap approaches the customer's body, the light of the screen will be turned off to save the power. Because proximity sensors can function without any physical touch, this sensor becomes the key component in our proposed "no-touch" mechanism for initializing applications on smart phones.

To understand the implementation and performance issues of our proposed "no-touch" mechanism, we apply the proposed "no-touch" mechanism to the speech-oriented applications on Android platform. The considered speech-oriented applications in our demonstration is speech recognition, which can transform voice to text and control the device to perform the corresponding tasks. In [2], the basic architecture of "speech recognition" was introduced, including the equipment of storing audio files, analog to digital module, the sampling audio file and matching algorithm. During the process of recognition, the recognition server will extract the features of input audio file, match the sampling in the database and find the most similar one to be the recognition result. Through speech recognition, users can do a lot of jobs with "hand-free" which is convenient for the users. In recent years, Google also launched a speech search on the handset. Users use their voice to order the handset searching on the Internet. The details of this application are in [3]. Therefore, the combination of smart phone and speech recognition becomes a modern trend. But users still need to touch screen to activate every speech-oriented application on smart phone. Thus, we implement proximity based "no-touch" mechanism on speech recognition to make speech-oriented application achieve real controlling by voice.

III. SYSTEM DESIGN

In this section, we introduce two main mechanisms to distinguish waving hands from taking the handset and enhance the speech recognition accuracy.

A. System design

The idea of designing the "no-touch" mechanism for smart phones is to replace the action of touching panels with some kinds of sensors to trigger the event. Owing to the attributes of power saving and insensitive to the surroundings, proximity sensors are selected to initiate speech recognition. When a user is close to the handset, proximity sensors can be triggered. Users can wave their hands on the top of the proximity sensor to initiate any application. Thus they need not watch their handsets to click small icons.

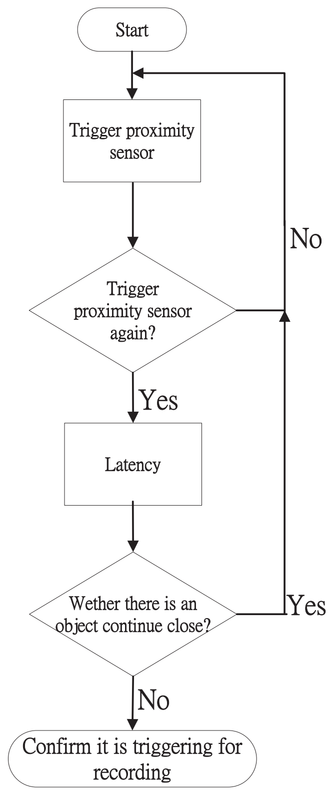

Fig. 1. The flow of waving hands testing mechanism (WTM).

However, that proximity sensors may be mis-triggered when users take the handsets. Therefore, we design a mechanism, called waving hands testing mechanism (WTM), to distinguish waving hands from taking handsets. On the other hand, the longer the proximity sensor is on, the more the power is consumed. In the speech oriented applications, proximity sensors are switched off after voice recording. In order to reduce the period of proximity sensor switching on, we design a "speech confirmation flow" that enhance the accuracy of recognition up to 94%. The following are the details of these two mechanisms.

B. Waving hands testing mechanism (WTM)

In WTM, users have to wave hands twice and wait for a short time latency. The purpose of waving hands twice can avoid users touching proximity sensors inadvertently. After a short latency, our mechanism will check whether any object is still approaching, which may imply that users are taking their handsets. Adopting WTM can reduce the possibilities of mis-triggering sensors. The flow chart of WTM is in Fig. 1.

C. Speech confirmation mechanism

Another issue in implementing proximity sensors driven speech recognition is that inaccurate detection costs a

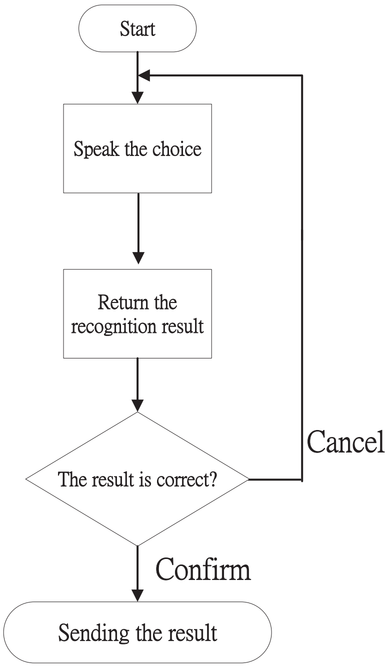

Fig. 2. The flow of speech confirmation.

great amount of resource in repeated communications and computation. In order to enhance the speech recognition accuracy, we design the recognition candidates, such as "1,2,3,", and we adopt speech confirmation mechanism to check the recognition result. If the result is wrong, the recognition process will be started again. The flow chart of speech confirmation is shown in Fig. 2.

WTM and speech confirmation mechanism can be implemented on any proximity based "no-touch" mechanism for speech oriented application on the handsets. These two mechanisms can help proximity based "no-touch" mechanism become more friendly and and convenient.

Fig. 3 shows that combining WTM and speech confirmation mechanism can be implemented on any speech oriented applications. In this figure, we only switch on proximity sensors between the periods of WTM executing, which can save the power of handsets. These procedures include the following four steps. (1) Start the application. (2) Start WTM. In WTM, proximity sensors sense whether the distance between the object and the proximity sensor is changed and decide the action of users. (3) As WTM decides the user's action is waving hands, the recorder is switched on to record and then the speech confirmation mechanism is activated. While the system executes the speech confirmation mechanism, the proximity sensor is switched off. (4)After the recognition result has been checked by users, the system will return the final result.

IV. DISCUSSION

The design of the proximity based "no-touch" mechanism for speech oriented application is implemented with the following features:

(1) Using proximity sensor to trigger speech recognition.

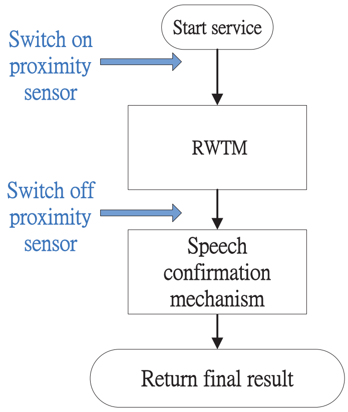

Fig. 3. The flow of using proximity sensor to activate speech recognition.

TABLE I

Power consumption of sensors

Sensor	Value(Power Consumption)
Orientation sensor	9.7mA
Magnetic field sensor	6.7mA
Accelerometer	3.0mA
Light sensor	0.5mA
Proximity sensor	0.5mA

We choose proximity sensors to trigger speech recognition because proximity sensors consume the least power on the handset and insensitive to the surroundings.

There are a lot of sensors on the handset, such as acceleration sensors, gravity sensors and light sensors, etc. Using light sensors and proximity sensors can make users operate the handset without focusing on the screen. The light sensors will be triggered when users go into a tunnel during driving. So light sensors are not suitable for playing this role. Table I shows the power consumption of sensors on HTC Desire. From Table I, we find that the proximity sensor consumes least power. Thus we select proximity sensors to trigger speech recognition. This implementation carries out the features of speech service "without using hand" and makes handset interface more user-friendly. We achieve power saving and convenient at the same time in this implementation.

(2) Users can determine when to trigger speech recognition.

In current speech oriented application, users need to reply immediately as the request arrives. Otherwise, the recognition will be failed. In proximity based "no-touch" mechanism on speech oriented applications, users can activate speech recognition by waving their hands whenever they are free.

(3) Adopting WTM and speech confirmation mechanism.

WTM uses repeated waving hands for a short period of time to reduce the probability of mistriggering the sensors, which also save the power from misacivation of

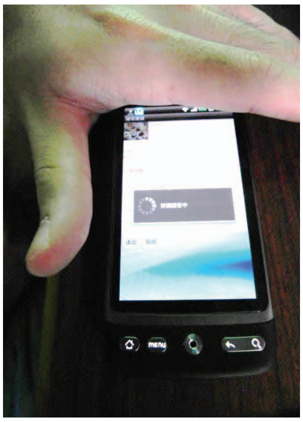

Fig. 4. System operation.

TABLE II
WTM performance result

Mechanism	Testing Times	Error Times	Error Rate
Without using WTM	**200 times**	**108 times**	54%
Using WTM	**200 times**	**2 times**	1%

TABLE III
Speech recognition performance result

Mechanism	Testing Times	Error Times	Error Rate
Google speech recognition	**100 times**	**43 times**	43%
The speech recognition of our system	**100 times**	**6 times**	6%

speech recognition. Besides, proximity sensors are only switched on during executing WTM to save more power compared to switching on proximity sensors all the time. In addition to WTM, we also design a speech confirmation flow to reduce recognition error rate. Setting recognition candidates and adopting recognition confirmation flow for users make recognition successful rate achieve 94%.

(4) Sensor frequency choosing advice.
"$SENSOR\_DEALY\_FASTEST$, $SENSOR\_DEALY\_GAME$,$SENSOR\_DEALY\_UI$ and $SENSOR\_DEALY\_NORMAL$" are four sensing frequencies on the handsets. If the highest frequency is set higher, the power of the handset will be consumed fast. It is critical to choose the suitable level of frequency and the time spot for switching the sensors to minimize the handset power consumption. In our experiments, we find that choosing $SENSOR\_DELAY\_NORMAL$ to be the sensing frequency can achieve the better successful rate and the lower power consumption. The details of experiment will be discussed in the next section.

V. PERFORMANCE EVALUATION

In this section, we show the experimental results aiming at evaluating the proposed recognition accuracy improving mechanism. Next, we describe experimental environments.

A. Experiment Setup

We use HTC Desire as 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. Fig. 4 is shown the system operation.

B. Experiment 1

This experiment shows the error rates when WTM is used and vice versa. If users take the handset and trigger the recording, it will be judge error. Otherwise, it will be correct. Table II shows that the error rate is 54% when WTM is not used, much higher than that when WTM is used. From this result, it is implied that a speech recognition system adopting WTM will improve accuracy by filtering unnecessary triggers.

C. Experiment 2

This experiment compares the recognition rate of our speech recognition with Google speech recognition. We use "1,2,3,4,5" to be the recognizing input. The designed system will repeatedly recognize the input speech until the result is correct. We only record the result of first recognition in this experiment. If recognition result is same with the recoding, it will be correct. Otherwise, it will be error.

Table III shows that the designed speech confirmation flow can reduce the recognition error rate greatly by predicting users' possible commands and executing recognition process on the cloud platform. Cloud-based speech recognition can match the input audio file with many audio samples stored on the cloud database, and execute these processes in parallel.

D. Experiment 3

Four frequencies of sensors on the handset are considered in our design: $SENSOR\_DEALY\_FASTEST$ (sensing interval is 0 ms), $SENSOR\_DEALY\_GAME$ (sensing interval is 20 ms), $SENSOR\_DEALY\_UI$ (sensing interval is 60 ms), $SENSOR\_DEALY\_NORMAL$ (sensing interval is 200 ms). Different sensing frequency will cause different power consumption. We set three sensing frequency which are $SENSOR\_DEALY\_FASTEST$, $SENSOR\_DEALY\_NORMAL$ and without using sensor in this experiment to observe the power consumption. We measure how much time the proximity sensor with different sensing frequency can be used if there is 1% power.

Table IV shows that power consumption of $SENSOR\_DEALY\_NORMAL$ mode and without using sensors. It is shown that the difference is not significant. The sensing

TABLE IV

Proximity sensor frequency performance result

Sensing Frequency	The time of proximity sensor can be used with 1% power	The successful rate of sensing waving hands
Fastest mode	**2 mins 30 seconds**	99%
Normal mode	**3 mins 23 seconds**	95%
Without using sensors	**3 mins 55 seconds**	

successful rate of $SENSOR\_DEALY\_FASTEST$ mode and $SENSOR\_DEALY\_NORMAL$ mode is very close. This is because the frequency of waving hands can not be so fast. Thus, the successful rate will not be influenced. That is, choosing $SENSOR\_DEALY\_NORMAL$ mode to be the sensing frequency can achieve the best performance.

VI. CONCLUSION

In this paper, we proposed a proximity-based "no-touch" mechanism to initiate mobile applications without the need of touching the screen. The proposed mechanism was implemented in Android phones for the application of speech recognition. To enhance the accuracy of event detection and power efficiency, "waving hands testing mechanism (WTM)" and "speech confirmation mechanism" were developed. The experiment results show that WTM can eliminate unnecessary triggering of speech recognition, and speech confirmation mechanism can achieve successful recognition rate up to 94%. Combining both mechanisms can switch proximity sensors on and off precisely, thereby saving power significantly. Last but not least, we found that how to select the most suitable parameter, such as sensing frequency $SENSOR\_DELAY\_NORMAL$, to lower power consumption and increase successful detection rate of proximity sensors. It is worthwhile further investigate how to exploit the potential of the proposed "no-touch" mechanism in other applications, such as emergency communications.

References

[1] Nils Christensson, "Apparatus and method of controlling a voice controlled operation," in *US Patent No.6,532,447*, 2003.

[2] Richard D. Peacocke and Daryl H. Graf, "An Introduction to Speech and Speaker Recognition," *IEEE Computer Society Press Los Alamitos*, 1990.

[3] http://www.google.com/mobile/voice-search/ .

[4] Ken Hinckley, Jeff Pierce, Mike Sinclair and Eric Horvitz, "Sensing Techniques for Mobile Interaction," *ACM Symposium on User interface software and technology*, 2000.

[5] Yu-Shuo Chang, Shih-Hao Hung, Nick Wang and Bor-Shen Lin, "CSR: A Cloud-assisted Speech Recognition Service for Personal Mobile Device," *IEEE International Conference on Parallel Processing(ICPP)*, 2011.