Hand-free Gesture Recognition for Vehicle Infotainment System Control

Qi Ye, Lanqing Yang, Guangtao Xue
Shanghai Jiao Tong University
{yeqi3614, yanglanqing, gt_xue}@sjtu.edu.cn

***Abstract*—Gesture recognition has a rapidly growing market size which is forcasted to increase from 14 billion in 2012 to 44 billion in 2020. Applying gesture recognition for vehicle infotainment system control is considered a promising alternative against traditional buttons, touch screens, or even speech-based control for its numerous advantages. However, existing gesture control solutions either depend on camera which imposes privacy concern and is sensitive to light condition or require users to wear a device on their hand which makes it inconvenient to use. Therefore, the work proposes to use the acoustic-based device-free hand tracking technology for gesture recognition. Because it only requires an ordinary speaker and microphone which are already available on vehicles or equipped in smart phones, it doesn't cast additional hardware cost or installation. We implement the proposed gesture control in Android phones and show it's feasibility for vehicle infotainment system control.**

***Index Terms*—Gesture recognition**

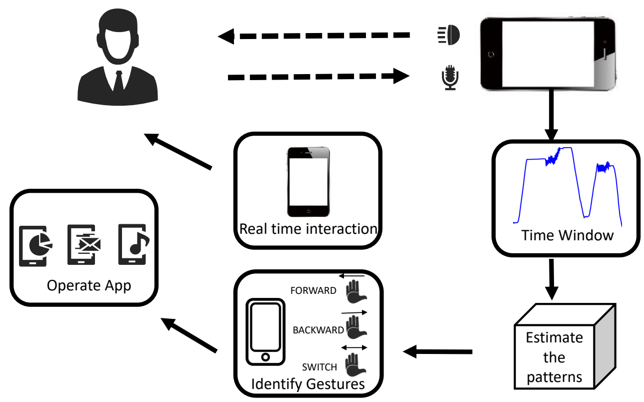


Fig. 1: System Overview

I. Introduction

A study from the AAA Foundation for Traffic Safety shows that the entertainment and information systems in many vehicles can distract drivers for more than 40 seconds at a time [5]. While removing eyes from the road for two seconds doubles the risk of an accident, spending 40 seconds to operate infotainment systems is extremely dangerous and considered as a huge threat for both the driver and all road users. Although some automakers address this distraction by disabling navigation function while the car is in motion, many vehicle models still allow it. With on-board features growing in numbers and becoming a bigger selling point for automakers, people are looking for better control methods over car infotainment systems.

With many things on the roadways demanding drivers' attention, such as errant motorists or speed limits, infotainment systems have to be designed to help drivers organize and operate all these various functions without piling on the stress. Nowadays, we can find a touchscreen fitted to most cars, ranging from low-end to the top-of-the-range vehicles. However, touch screens and buttons based navigation is still time intensive and requires drivers to take their eyes off the road. On the other hand, with the increasing popularity of voice control speakers, like Amazon Echo and Google Home, some vehicles also embrace the voice recognition technology[1]. However, voice recognition is known to be frustrating for its recognition accuracy, especially under noisy environment in vehicles.

In this paper, we propose to apply gesture recognition for vehicle infotainment system control. Gesture recognition is considered as a promising alternative for its intuitive usage and customizable input. Unlike touch screens and buttons with which a driver needs to know exactly where to touch or click to perform corresponding actions, gesture recognition allows a driver to perform actions anywhere in the air. Therefore, the driver can keep eyes on the road. Our APP provides customizable functions to map various gestures to the predefined operation in cars.

Existing gesture control solutions either depend on camera which imposes privacy concern and is sensitive to light condition or require users to wear a device on their hand which makes it inconvenient to use [4], [3], [5]. Therefore, we propose to use the acoustic-based device-free hand tracking technology for gesture recognition adapted from Mobil-Trakk motion tracking SDK provided by Hauoli LLC [2]. Because Mobil-Trakk SDK only requires an ordinary speaker and microphone which are already available on vehicles or smart phones, it doesn't cast additional hardware cost or installation. We implement the proposed gesture control in Android phones and show it's feasibility for vehicle multimedia system control.

II. System

We first describe how our technology works as an APP installed in a smart phone with a built-in speaker and microphone. The system flow is shown in Fig. 1 and each step is detailed as follows:

1D Distance Estimation: The speaker of the mobile phone continuously emit high-frequency (i.e., 17-20KHz) sound signals which are inaudible to human. When these emitted signals hit surrounding objects, they are reflected back and captured by the microphone. Each reflected signals traveled through different paths in the vehicle for corresponding distances. Our goal is to track the movement of the hand for gesture recognition. In order to distinguish the signals reflected by the hand movement from those reflected by surround objects, we use the Mobil-Trakk Library [2] to only monitor the object with largest movement. The update rate of distance estimation is $48KHz$ and we down-sample it to $25Hz$ to reduce computation time and energy.

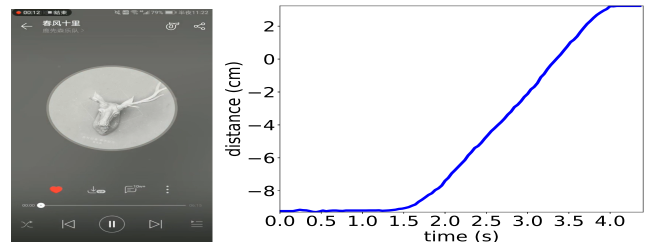
Gesture Recognition: We identify the gesture of the current user by analyzing the distance estimation retrieved above. We maintain a sliding window to continuously detect if any gesture is performed in the current window. There are two important design principles for our gesture recognition algorithm. First. we want to allow drivers to perform gesture at any place in the vehicle so they can keep eyes on the road so the absolute distance between the hand and the phone can be different for various drivers and vary each time. Moreover, the estimated distance could suffer from the error accumulation issue, the absolute distance is not reliable. Therefore, to make the recognition robust, we only rely on the change in distances during each window. The second principle is that the gestures need to be as simple as possible so drivers can concentrate on driving. On the other hand, to avoid the false alarm, we need to make gestures distinct from each other as well as from casual hand movements.

With the above two design principles in mind, we design several gestures and perform survey to get feedback. In the end, we currently identify three gestures for initial experiments. When the hand moving distance exceeds a threshold and the distance monotonically increases in a window, the current action is recognized as $BACKWARD$. When the distance exceeds a threshold while the distance monotonically decrease in a time window, the gesture is recognized as $FORWARD$. When the hand move back-and-forth quickly in a time window, the gesture is recognized as $SWITCH$.

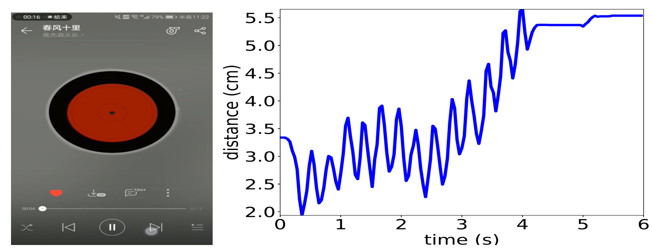
Note that the size of the window size on one hand determines the latency to recognize a gesture and on the other hand the robustness of recognition. In other words, when the window size is small, recognition delay is shorter and user may feel the APP is more responsive; however, smaller size also suggests that false alarm is more likely to happen. To balance the trade-off, we carefully perform extensive experiments and set the windows size to $3s$.

Customized Action: In our APP, we allow users to map each gesture to corresponding actions, for example, play music, read twitter, open navigation app, and etc. The customized mapping is achieved by applying the Android publish-subscribe design pattern where we define broadcast messages from/to the Android system and other Android Apps. Broadcasts are sent when corresponding gestures are detected. Apps (actions) of interest filter out broadcast messages and only react to the desired gestures.

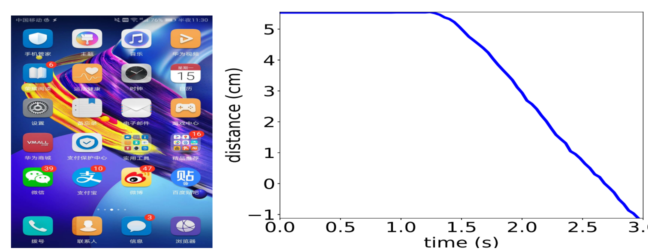
For example, in Fig. 2 shows that we map $FORWARD$ to



(a) $FORWARD$ and open music player.



(b) $SWITCH$ and play next song.



(c) $SWITCH$ and play next song.

Fig. 2: The screen-shot of our APP.

start a music player App. When $SWITCH$ is detected, the player play the next song in the list. When $BACKWARD$ occurs, the player stops.

References

[1] Amazon-vs-google. Available online: https://thewirecutter.com/reviews/amazon-echo-vs-google-home/.

[2] Hauoli llc. Available online: https://www.hauoli.co/.

[3] J. Kela, P. Korpipää, J. Mäntyjärvi, S. Kallio, G. Savino, L. Jozzo, and D. Marca. Accelerometer-based gesture control for a design environment. *Personal and Ubiquitous Computing*, 10(5):285–299, 2006.

[4] P. A. Nilsson. Camera based control, May 18 2010. US Patent 7,721,207.

[5] D. L. Strayer, J. M. Cooper, R. M. Goethe, M. M. McCarty, D. Getty, and F. Biondi. Visual and cognitive demands of using in-vehicle infotainment systems. 2017.