Contactless Gesture Recognition System Using Proximity Sensors

Heng-Tze Cheng*†, An Mei Chen†, Ashu Razdan†, Elliot Buller†
*Carnegie Mellon University †Qualcomm Incorporated

***Abstract*—In this paper, we present a novel contactless gesture recognition system using proximity sensors. A set of infrared signal feature extraction methods and a decision-tree-based gesture classifier are proposed. The system allows a user to interact with mobile devices using intuitive gestures, without touching the screen or wearing/holding any additional device. Evaluation results show that the system is low-power, and able to recognize 3D gestures with over 98% precision in real time.**

I. Introduction

Gesture-based interfaces provide an intuitive way for users to specify commands and interact with computers [1]. Existing gesture recognition systems can be classified into three types: *motion-based*, *touch-based*, and *vision-based* systems. For motion-based systems [2], [3], a user must hold a mobile device or an external controller to make gestures. Touch-based systems [4], [5] can accurately map the finger/pen positions and moving directions on the touch-screen to different commands. However, 3D gestures are not supported because all possible gestures are confined within the 2D screen surface. While the first two types of system require users to make contact with devices, vision-based systems [1], [6] using camera and computer vision techniques allow users to make intuitive gestures without touching the device. However, vision-based systems are computationally expensive and power-consuming, which are undesirable for resource-limited mobile devices like tablets or mobile phones.

To solve the existing challenges, we present a novel gesture recognition system with the following contributions:

- The design and evaluation of the first contactless gesture recognition system using only infrared proximity sensors.
- The proposed infrared (IR) feature set and classifier for real-time 3D gesture classification.
- Reducing the power consumption of gesture recognition.

The design also reduces the frequency of users' contact with devices, alleviating the wear and tear of screen surface.

II. System Design and Methods

A. Proximity Sensor Data Acquisition

For the configuration under study, a proximity sensor consists of two IR LEDs and a IR receiver (see Fig. 1), which are placed underneath a plastic/glass screen surface, surrounded by optical barriers. The LEDs emit IR strobes in turns as two seperate channels. When a hand or any object is near, the receiver detects the reflection of the IR light, whose intensity increases as the object distance decreases. The light intensities of the two IR channels are sampled by the firmware at 100Hz.

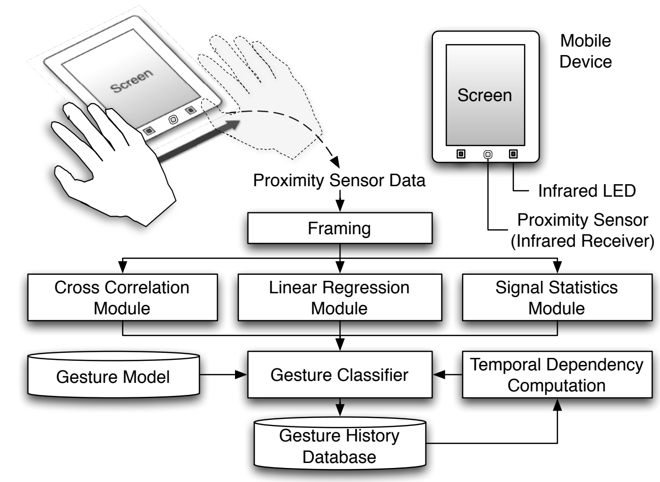


Fig. 1: The architecture of the gesture recognition system.

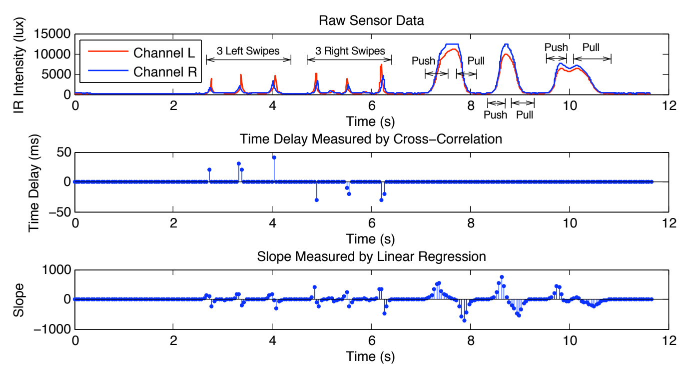


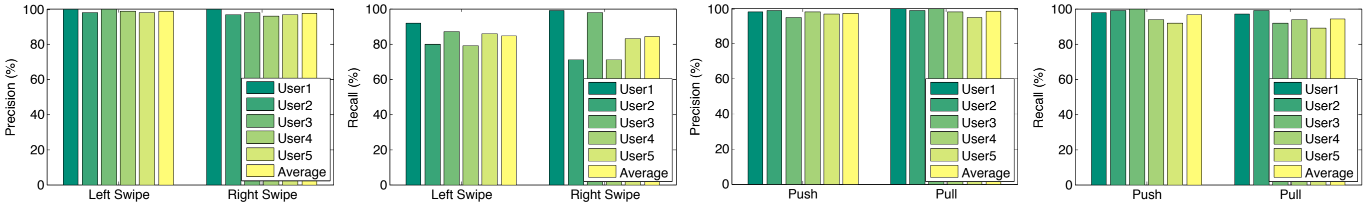
Fig. 2: An example of proximity sensor data and IR features.

B. Gesture Recognition Algorithm

The algorithm continuously scans the input IR intensity data and decides if a predefined gesture is observed. First, the data is divided into 50% overlapping frames, 140 ms each. Then, three types of feature are extracted from each frame:

1) Inter-channel Time Delay: The feature measures the pair-wise time delay between the sensor data of two channels, which shows how a hand approaches the IR LEDs at different instants. This corresponds to different moving directions of hands (see Fig. 2 for example). The time delay t_D is calculated by finding the time shift n that yields maximum cross correlation value of two discrete signal sequences f and g:

$$t_D = \arg\max_n \sum_{m=-\infty}^{\infty} f^*(m)g(m+n) \tag{1}$$



(a) Precision of left/right swipe (b) Recall of left/right swipe (c) Precision of push/pull (d) Recall of push/pull

Fig. 4: Precision and recall rate of gesture recogntion.

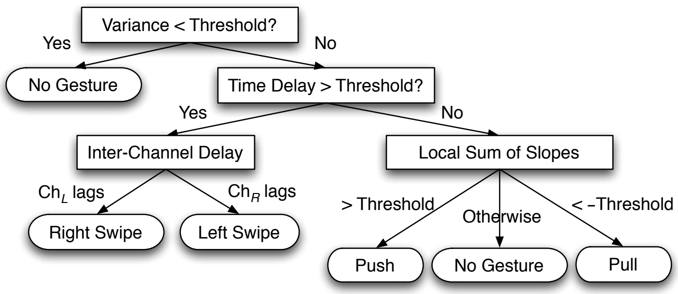


Fig. 3: Illustration of decision-tree-based gesture classifier.

2) Local Sum of Slopes: This feature estimates the local slope of the signal segment within a frame, which shows how fast a user's hand is moving toward or away from the proximity sensors. The slope is calculated by first-order linear regression, and then summed up with the slopes of the 6 previous frames. The local sum better captures the continuous trend of slopes rather than sudden changes.

3) Signal Statistics: The mean and variance of raw data in the current frame and the history of previous frames.

After feature extraction, a decision-tree classifier shown in Fig. 3 is designed to classify the frame as one of the gesture in the predefined gesture model, or report that no gesture is detected. We also keep a history of 7 frames to take temporal dependency between consecutive frames into consideration. For example, when a gesture is detected, the system suppresses the output of the same gesture for 6 frames because it is hard for a user to make the same gesture again very quickly. Once the gesture sequence history of a user is obtained, the transition probability between gestures can also be incorporated to improve the recognition accuracy.

We implemented the system using Silicon Labs Si1120 infrared proximity sensor [7]. The gesture recognition algorithm was implemented in C++. The frame sizes and thresholds are empirically set to minimize false alarm through experiments.

III. Evaluation

We evaluate the system on four most common gestures: *left swipe*, *right swipe*, *push* (hand moving vertically down toward the device), and *pull* (hand moving vertically up away from the device). The system is evaluated on a gesture dataset collected from 5 users, including 1 left-handed and 4 right-handed user. The dataset consists of 2,000 gesture samples in total, with each user performing each gesture 100 times. To prevent users from adapting to the system over time, the recognition results were not exposed to the users during the data collection.

1) Recognition Performance: We use the widely used precision/recall metric to evaluate the performance:

$$precision = \frac{TP}{TP + FP} \qquad recall = \frac{TP}{TP + FN} \tag{2}$$

where TP, FP, FN refer to true positive, false positive, and false negative, respectively. As shown in Fig. 4, the system achieved 98% precision in average, and is robust from user to user. The high precision implies low false alarm rate, which is ideal for gesture recognition because executing a wrong command is usually worse than missing a command. The recall rate is 88% in average, which is lower than precision because the system can miss gestures when the hand is too far from the sensor, or when a gesture is performed much slower than usual.

2) Power Consumption: The system power is dominated by the power consumed by IR LED and the control chip:

$$P_{LED} + P_{chip} = f_{conv} \cdot T_{prx} \cdot (I_{LED} + I_{chip}) \cdot V_{LED} \tag{3}$$

which is only 0.3 mW (idle) to 20 mW (active, with larger T_{prx} when an object is in proximity) [7], much lower than the 200-mW power budget for typical UI of mobile device [8] (f_{conv} and T_{prx} are conversion frequency and pulse width).

IV. Conclusion

We have presented a contactless gesture recognition system that allows users to make gesture inputs without touching, holding, or wearing any device. Using the proposed IR feature set and classifier, the system can recognize 3D gestures with 98% precision. The low power consumption and high recognition accuracy make the system particularly desirable for deployment on resource-limited mobile consumer devices.

References

[1] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 677–695, 1997.

[2] A. Wilson and S. Shafer, "XWand: UI for intelligent spaces," in *Proc. SIGCHI conf. Human factors in comput. syst.*, 2003, pp. 545–552.

[3] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uWave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive Mob. Comput.*, vol. 5, no. 6, pp. 657–675, 2009.

[4] W. C. Westerman and J. G. Elias, "System and method for packing multi-touch gestures onto a hand," Patent 7 030 861, April, 2006.

[5] J. O. Wobbrock, A. D. Wilson, and Y. Li, "Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes," in *Proc. ACM Symp. User interface software and technology*, 2007, pp. 159–168.

[6] M. H. Yang, N. Ahuja, and M. Tabb, "Extraction of 2D motion trajectories and its application to hand gesture recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1061–1074, 2002.

[7] Silicon Labs, *Proximity/ambient light sensor with PWM output*, 2009.

[8] Y. Neuvo., "Cellular phones as embedded systems." in *IEEE International Solid-State Circuits Conference*, 2004.