SoundSense: 3D Gesture Sensing using Ultrasound on Mobile Devices

Yen-Ting Liu[1], Che-Yang Wu[2], Pi-Hsun Shih[1], Hsiang-Sheng Liang[1], Mike Y. Chen[2]
[1]Department of Electrical Engineering, National Taiwan University
[2]Department of Computer Science and Information Engineering, National Taiwan University

ABSTRACT

We present SoundSense, an approach to detect 3D gestures using ultrasonic rangefinders. Compared to other computer vision and proximity sensors techniques, our method consumes less power, which is suitable for mobiles devices, while retaining 82.2% cross-validation accuracy. Through a 16-participant design study, we identified the average operating distance from both the user's hand and face to the smartphone or tablet when performing mid-air gestures. We compared machine learning algorithms SVM and HMM for recognition accuracy. We have implemented SoundSense, which works in real-time on Android tablet. Qualitative feedback is also presented from a 6-participants usability study.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Input devices and strategies.

General terms: Design, Human Factors

Keywords: Ultrasound, rangefinders, 3D gesture, bare-hand gestures

INTRODUCTION

On modern mobile devices like smartphones and tablets, the capacitive touch screen is the dominant input method. However, there are situations when it is not convenient for the user to use direct touch input. For example, when the user is cooking or eating, she or he wouldn't want to touch the screen with greasy hands. In other cases, it is handy to control the device when it is in a distance away. A user might want to shut down the alarm immediately while he/she is away from the device. The problem would be solved if we can manipulate the device by means other than touch input.

Computer vision and speech input are available options. Techniques have been developed for tracking hand gesture by camera videos. Nevertheless, computer vision methods are often vulnerable to lighting condition and also require larger power consumption, which is less suitable for mobile devices. Voice input enabled by speech recognition is available on devices like Apple iPhone 4S. However, it is vulnerable to the noisy environment.

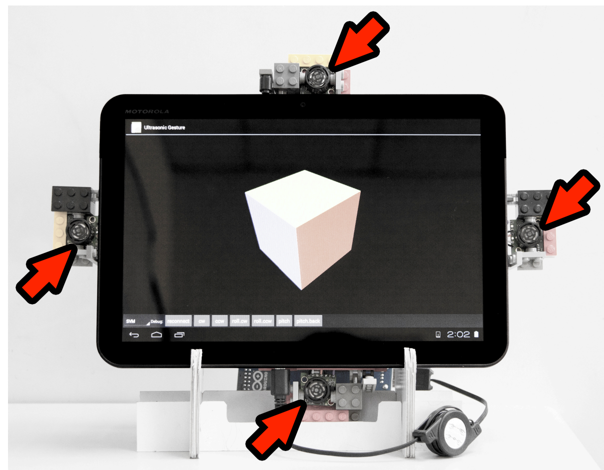

Figure 1: SoundSense Prototype consists of four ultrasonic rangefinders (pointed by the arrows) and an Android tablet. The application in the screen is used in the usability test.

We present SoundSense, which use ultrasonic sensing techniques to detect above-screen bare-hand gestures. Not only the user can manipulate the mobile devices without actually touching the screen, but also no additional gadget is required on the user's hands. Ultrasound sensing technique is immune to lighting condition and noisy environment. It is also more power efficient than other hardware components like camera and proximity sensor. (See Table 1)

While our objective is to detect above-screen gesture, the resolution requirements may not be as high as tasks like precise positioning or selection. A set of 12 gestures is designed, presenting 6 degree-of-freedom in 3D space. Figure 1 presents the SoundSense prototype. The 4 ultrasonic rangefinders mounted on the Android tablet device provide 4 distance values. Comparing to the method of Doppler ultrasonic sonar, rangefinders provides the actual distance in addition to the movements.

It is a problem to recognize the beginning and the end of a 3D gesture. The problem is also referred as the clutching problem. A trigger algorithm is designed to mark the beginning of the gesture: the users are asked to cover an arbitrary sensor for a short period before a gesture movement.

	Ultrasonic Rangefinder	Proximity Sensor	Computer Vision (single camera)	Depth camera (Kinect)
Cost	$3.15 * 4 = $12.6 (*)	$8 * 4	No addition cost	Approx. $56 (*)
Power consumption	8.82 * 4 mW	23.2 * 4 mW	About 185 mW	12.96 W
Working range	10cm ~ 600cm	5cm ~ 20cm [GW]	Depends on algorithm and camera resolution	1.2m ~ 3.5m
Vulnerable to lighting condition	NO	NO	YES	YES (IR interference)
Distance information	YES	NO	NO	YES
Sampling rate	20 Hz	125 Hz	30 fps	30 fps

Table 1: Comparison between methods of bare-hand gesture recognition on mobile devices (*): bill-of-material

Once the beginning of a gesture is identified, the machine learning algorithms is capable of analyzing the time sequence of the distance values. The recognition performance and the usability test indicate that SoundSense is a feasible solution for ultrasonic gesture recognition.

Our paper makes the following contributions: 1) using ultrasonic rangefinders to recognize 3D bare-hand gestures on mobile devices, 2) identify the operating distance of 3D gestures on smartphone and tablet, 3) distance threshold is used as gesture delimiter to solve the clutching problem, and 4) evaluation of 2 3D gestures recognition implementations (HMM, SVM).

RELATED WORKS

Our work relates to previous work in two categories: First, approaches of above-screen bare-hand gesture recognition on mobile devices are presented and compared. Second, there are works that combined ultrasonic sensing techniques with mobile devices, by using built-in speakers and microphones as well as external sensors.

Ultrasound Gesture

Kreczmer [9] proposed 4 bare-hand gestures to interact with a robot which are recognized by ultrasonic rangefinders. “Approach me” and “go away” gestures are linear hand movements toward and away from the ultrasonic rangefinders. “Turn left” and “turn right” gestures are horizontal linear hand movements. Both two categories of gestures are recognized by heuristic algorithm based on the speed and the direction of the hand movement.

Gupta et al. [6] leveraged built-in speaker and microphone in commodity devices to sense in-air gesture. Inaudible tone was generated, which gets frequency-shifted when it reflects off moving objects like the hand due to the Doppler effect. Such shift was measured with the microphone to infer 5 gestures.

3D gestures

Other options to recognize the above-screen bare-hand gestures include computer vision and proximity sensors. Microsoft Kinect features a combination of RGB camera and depth sensor to provide 3D motion tracking that includes bare-hand gestures. Mariappan et al. [12] implemented a computer vision engine that are capable of hand tracking on Android mobile devices by using the built-in camera of the smartphone.

Kim et al. [8] utilized four proximity sensors onto a wrist watch, arranged to facing up in a cross shape, to sense hovering gestures above the watch. 9 Gestures around the wrist watch is designed, consisting of combinations of passing the area above the watch forward and backward for one or two times, changing directions when above the watch, covering all sensors and rolling around the watch. The digital high/low data collected by the four proximity sensors are passed to GART [10] HMM recognition model via Bluetooth, with the calculation carried out in a remote computer. A comparison of above methods is shown in Table 1.

A challenge of above-screen gesture is to distinguish intentional movements from other noise. Choosing a gesture delimiter to mark the start and end of a movement is a solution. Good gesture delimiter design should prevent natural human movements, like clutching, from triggering the gesture recognition. The delimiter itself should remain simple and easy to perform [2]. Gestures recognized by computer vision algorithms often use pinch [2,15] or grasp [14] as the delimiter. Others may design gestures that are essentially clutch-free [11]. Visual feedbacks are shown after recognizing the gesture delimiter to inform the user.

Ultrasound and Mobile devices

Borriello et al. [4] combine wireless networking and microphone interface of a mobile device to perform room-level positioning. Wireless data and ultrasound pulses were generated from the PCs in the room. A PDA carried by the user listened to both signal and calculated the indoor position according to it. There was no special hardware required.

Arentz et al. [1] achieved short range directional data-communication on a smartphone using built-in speakers and microphones. The digital signals are encoded into ultrasound with different pulse width. The data transmission rate is limited by the A/D and D/A sampling rate. Bihler et al. [3] used ultrasonic signals which sent by a cheap, standalone emitter for indoor positioning. The built-in microphone of smartphone was the basis of the proximity detection system.

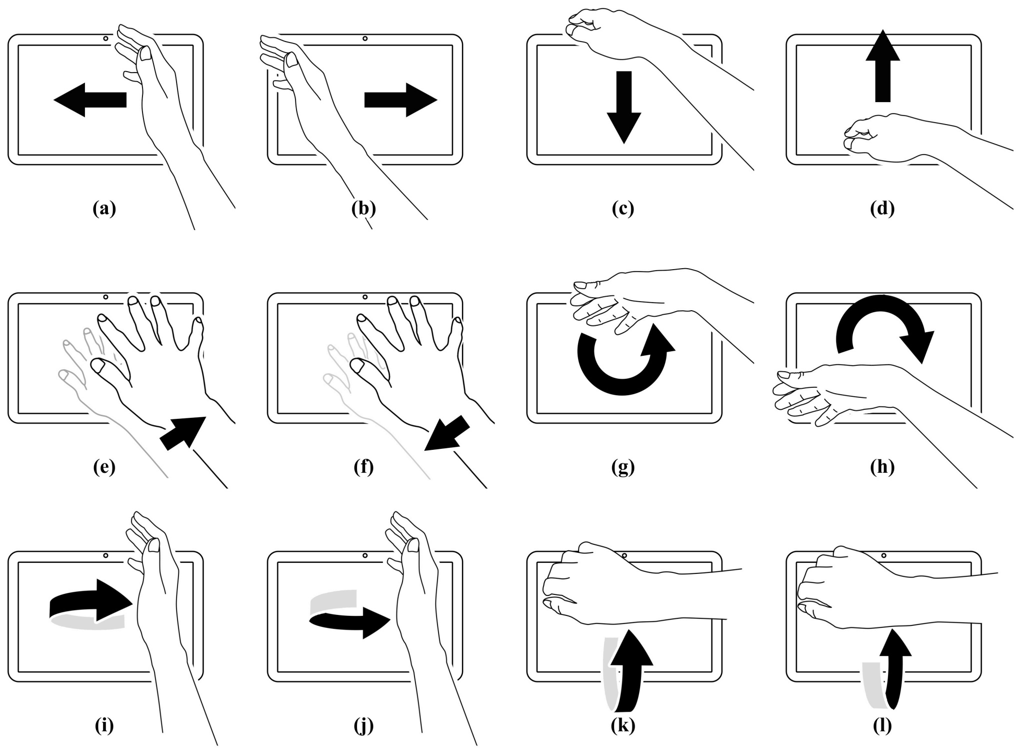

Figure 2: 12 Gestures of SoundSense. (a)-(d) are linear movements on 3 axes: R2L (right-to-left), L2R (left-to-right), T2B (top-to-bottom), B2T (bottom-to-top), N2F (near-to-far), F2N (far-to-near). SoundSense also supports 3D rotation gestures (g)-(l): CCW (counter-clockwise), CW (clockwise), RCW (roll-clockwise), RCCW (roll-counter-clockwise), PCW (pitch-clockwise), PCCW (pitch-counter-clockwise).

GESTURE DESIGN

We designed 12 3D bard-hand above-screen gestures, which is shown in Figure 2. There are 2 categories of gestures. The first group (a~f of figure 2) includes linear movements on 3 axes, namely R2L, L2R, T2B, B2T, N2F, F2N. The second group consists of rotation on xy, yz, xz planes, namely CW, CCW, RCW, RCCW, PCW, and PCCW. All these gestures form a 6 degree-of-freedom 3D interaction. As traditional 2D touch screen gestures like R2L, L2R are still available in 3D version, they are also extended to the third dimension.

FIRST APPRAOCH

In the first approach, we tried to build a ultrasonic range finder by the native speaker and microphone of a smartphone. Most modern mobile devices are equipped with speakers and microphones. Although they were designed for audible sound, it is also available for ultrasound. For a A/D converter sampling rate 44 kHz device, it is capable for processing 22 kHz signals according to Nyquist-Shannon sampling theorem.

The approach is suitable for proximity detection, but not available for range finding. We used 21 kHz as the frequency of the ultrasonic signal. The signal of the specific frequency is extracted by Gortzel algorithm. A pulse of X microseconds ultrasound pulse is sent and the microphone detects the reflection. In theory, we can calculate the time-of-flight of the ultrasound by measuring the time difference. However, the sampling rate of mobile devices is not enough to provide the required precision. In order to detect an object 17 cm above the screen, the time difference is 1 millisecond for the speed of sound is 340 m/s. There are only 44 samples in the period, which is much smaller than the 256-sample window of frequency analysis and leads to large error. We also tried to make use of the amplitude of the reflected signal, but the amplitude is so small that only proximity detection can be satisfied. To conclude, the low amplitude of the ultrasound signal and the low sampling rate of mobile device make it difficult to build an ultrasonic rangefinder with native hardware.

SECOND APPROACH

Additional ultrasonic rangefinders are used in the second approach. 4 MB1010-LV-MaxSonar-EZ1 ultrasonic rangefinders from MaxBotix provide the distance information about the objects above the screen surface. Each of them is placed on the center of the edges of the Motorola Xoom. These sensors are configured in simultaneous operation mode to avoid possible inference within sensors. Arduino Mega ADK first collect data then transfer to Android tablet after preprocessed. Every rangefinder reports the distance in the frequency of 20 Hz and forms a 4-vector. Typical data is like (198, 187, 20, 32) with the unit in centimeter.

By analyzing the sequence of such data we can recognize the user gestures.

We found that it is necessary to identify foreground signals from background signals. An ultrasonic rangefinder is able to detect object from 0 - 600 cm. While it is fully capable of detecting bare-hand gestures in foreground, other signals in background such as people walking from behind may also appear in the readings of the sensors. In this case, background signals are unwanted and inferences to gesture recognizing.

Design Study

The goal of the design study is to provide guidelines for gesture recognizing. Actual user data were gathered while the users were performing bare-hand above-screen gestures. 3 sets of data were collected: 1) the distance from hand to device, 2) the distance from face to device, and 3) the duration of the gestures. The measurement of the distance is shown in Figure 3. iPad 2 and iPhone 4 were used in the study, representing smartphone and tablet of screen size 3.5" and 9.7", respectively.

Experiment. These data are used to distinguish foreground gesture events from the background noise. The distance between screen and hand provides the information of possible gesture signal. Meanwhile, the distance between screen and face forms a lower bound of background signals. In addition, the duration time of gestures is in study for the parameter of gesture recognizing.

We hadn't implemented a functional prototype in this stage; the *proxy* method was used to simulate the bare-hand gesture interaction. Remote control software (Veency) was installed in both iPad and iPhone so that we can control the devices from our laptops. When the study participant performs a gesture, we make the correspond action on our computer. By telling the participants that we are talking notes for the study, the user believed that they were able to control the iPhone and iPad with bare-hand above-screen gestures.

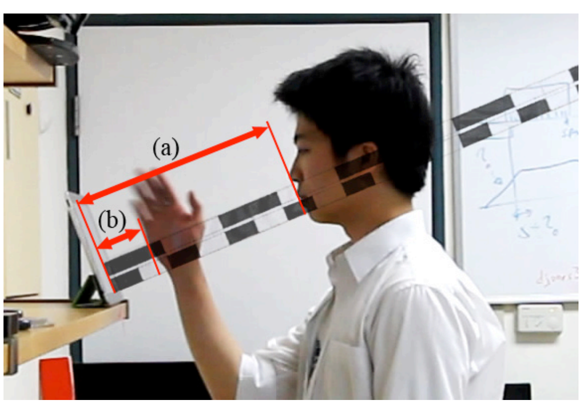

Figure 3: Captured image of design study. (a) and (b) are device-to-face and device-to-hand distance, respectively. The ruler image is shot at the exact position and camera settings. It is composited in Photoshop in order to get the distance readings. One black-and-white bar on the lower side of the ruler is 10 centimeters long.

The device was put on a rack, which is 135 centimeters above the ground. The user was standing throughout the experiments, so that the user could move to a conformable distance easily. Also, the participants were asked to step back from the device after finishing a single task. Therefore, the user was forced to move again and prevent the inference from the previous task.

Participants were asked to perform both text reading and photograph browsing. The font size and line height for text reading task is 16px / 28px for iPhone and iPad, respectively, which is the same as the iBook app. Full-screen pictures are shown in the picture browsing. We believe that the text-reading task may require the shortest distance while the picture browsing may require the longest distance. The text reading task utilized B2T and T2B gestures for page-up and page-down. R2L and L2R gestures are used in photograph browsing. Each participant performed one trial using an iPhone 4 and one trial using an iPad 2. The order of the devices and the tasks were both counter balanced across participants.

We recruited 16 participants (12 male, 4 female), from our university population. The participants' age ranged from 20 to 36. The average height of the participants is 172 centimeters. All of them have the experience with touch screen gestures.

Study Result. Figure 4 and 5 shows the histogram of the distance between the device to hand and to face for both iPhone and iPad. Only the average distance of each person is taken into account. The data of two tasks (reading and photo browsing) were combined together in the graph. The average distance from iPhone to hand and to face is 3.35 cm (SD 2.69 cm) and 27.82 cm (SD 7.42 cm), respectively. For iPad, the average distance to hand and to face is 4.30 cm (SD 3.53 cm) and 31.10 cm (SD 6.76 cm), respectively. As we can see in the figure, there is a valley at about 20 cm so that we can split the foreground signal with background signal by setting up a threshold. The average duration of gestures is 0.4 seconds, providing a guideline for actual recognition.

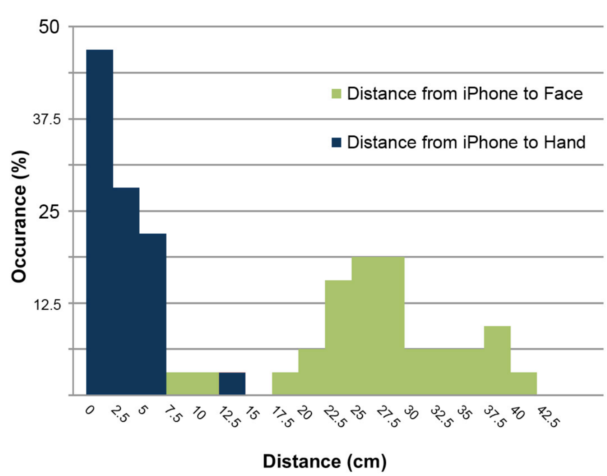

Figure 4: Histogram of user distance to the iPhone device.

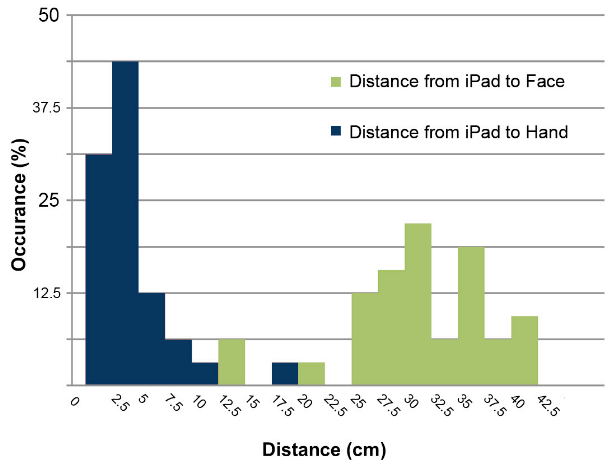

Figure 5: Histogram of user distance to the iPad device.

During the experiment with iPhone, some of the participants complained about the screen size of iPhone are hardly readable. They pointed out that they would not use iPhone from such distance, and the hand obscures the screen when doing gestures. IPad, on the other hand, does not have such problem. Therefore, we decided to extend our work on tablets instead of phone devices.

We decided to place 4 sensors on the middle of each edge of the device. Though the number of sensors should be as least as possible, it is required that the 12 gestures should be efficiently recognized. We noticed that out 5 of our 16 participants wave their hands only in small angle. Starting from the center, their hands only passes one of the 4 edges when doing gesture R2L and L2R. Thus, 4 is the minimum number if we want to detect movements from every possible direction.

Gesture Delimiter

Proximity distance information is used as the gesture delimiter to indicate the beginning of a gesture. Before each gesture starts, we count the number of frames the distance value being lower than the threshold. The threshold value is determined by the design study above, which is 20 cm. A gesture is marked as started if a single sensor is under the threshold for 3 consecutive frames (150 milliseconds). Once a gesture is started, a time window of 30 frames (1.5 seconds) is recorded for training and prediction. While a short time window is desired for faster recognition time, the number of the frames is determined by the gesture recognition method introduced below to provide higher accuracy. These parameters are chosen to be as unobtrusive as possible to the user, while maintaining the robustness of the gesture delimiter. This empirical solution also makes use of the depth information provided by the rangefinders, which is not possible with proximity sensors or the Doppler effect method.

Gesture Recognize Method

Two methods for gesture recognizing were implemented and compared: hidden Markov model (HMM), and support vector machine (SVM). We chose to utilize HMM given past success in previous work using them to model complex time series data [8]. SVM as a popular machine learning algorithm is implemented and compared with the HMM method.

The same training data is used for both SVM and HMM method. We recruited 16 participants (12 male, 4 female), from our university population. The participants' age ranged from 21 to 26. The average height of the participants is 170.5 centimeters. 3 participants are left-handed. The participants are demonstrated for the trigger method and the movements of all 12 gestures. A single trial consists of 12 gestures, and each participants were ask to record 2 trials of training data. A total of 32 training data were recorded for each gesture.

HMM. The Java library jahmm[1] is used for our HMM approach. 12 gestures are trained into 12 separate models. Each model consists of 10 states. Since we consider first order finite difference on values of 4 sensors, there are totally 8 variables in each observation and they are represented by a 8-dimensional multivariate Gaussian distribution. The first 4 dimensions are the sensor values of the current time frame, while the last 4 dimensions are the difference with the previous frame to take inter-frame relations into account. K-Means algorithm, which is provided by jahmm, is used to generate a rough model by clustering on the distribution of data. After that, we use Balm-Welch algorithm to iterate 100 times for training and improve the model.

SVM. We used Dynamic Time-Alignment Kernel (DTAK) [13] in LIBSVM [5] for our SVM approach. DTAK combined with radial basis function (RBF) enables non-linear time alignment for time series data, which is suitable for our application. 12 gestures are trained into a single multi-class SVM model. Each single sequential record is a vector with variant length, where each element is a four-dimensional vector that corresponds with four sensors value at the time frame. For the multi-class classification, though libsvm uses one-against-one method, we used DAGSVM [7] to reduce the computational complexity for mobile devices and still get good precision. We chose $C = 2$, $\gamma = 0.00048828125$ for LIBSVM parameter based on our experiment result, which was performed by a grid search with 5-fold cross-validation.

Performance

The recognition accuracy of HMM and SVM is 61.9% and 82.2%, respectively. The confusion matrix of HMM and SVM are shown in Table 2 and Table 3. Both results are conducted from a 5-fold cross-validation. The rows show the gesture the user was instructed to perform and columns show the classified result. The not available (n/a) indicates there are no or more than one gestures was recognized during the test period.

[1] http://www.run.montefiore.ulg.ac.be/~francois/software/jahmm/

	U2D	D2U	L2R	N2F	R2L	F2N	CW	CCW	RCW	RCCW	PCW	PCCW	n/a
U2D	22	1	0	1	0	0	2	2	0	0	3	1	0
D2U	0	23	0	0	0	0	1	1	0	0	0	7	0
L2R	0	0	19	0	0	0	6	4	2	1	0	0	0
N2F	2	2	0	12	0	10	0	0	2	0	1	3	0
R2L	0	0	0	0	20	0	3	2	5	1	0	1	0
F2N	1	1	0	6	0	12	0	0	1	1	7	2	1
CW	0	3	1	0	0	0	15	13	0	0	0	0	0
CCW	0	1	0	0	1	0	7	16	2	3	1	1	0
RCW	0	0	0	0	1	0	3	2	23	2	1	0	0
RCCW	0	0	1	0	2	0	0	0	4	24	1	0	0
PCW	0	1	0	0	0	0	0	0	1	0	27	3	0
PCCW	0	4	0	1	0	0	0	1	0	0	2	24	0

Table 2: Confusion matrix of HMM method.

	U2D	D2U	L2R	N2F	R2L	F2N	CW	CCW	RCW	RCCW	PCW	PCCW	n/a
U2D	32	0	0	0	0	0	0	0	0	0	0	0	0
D2U	0	27	0	1	0	0	0	0	0	0	0	4	0
L2R	0	0	28	0	0	0	3	1	0	0	0	0	0
N2F	0	1	0	27	0	0	0	0	0	2	1	1	0
R2L	0	0	0	0	27	0	2	0	3	0	0	0	0
F2N	0	0	0	2	0	25	0	0	0	2	0	2	1
CW	0	0	1	1	0	0	26	0	1	1	1	1	0
CCW	0	2	0	0	1	0	1	24	0	3	1	0	0
RCW	0	0	0	0	1	0	3	0	25	1	2	0	0
RCCW	0	0	1	0	0	1	0	1	1	27	0	1	0
PCW	0	1	0	1	0	2	0	0	0	0	24	4	0
PCCW	0	3	0	2	0	2	0	0	0	0	2	23	0

Table 3: Confusion matrix of SVM method.

The 4 simple gestures U2D, D2U, L2R, R2L have a higher accuracy than other gestures. RCW, RCCW, PCW, PCCW are harder to recognize since they all consists of z-axis movement as well as movements on x- or y-axis. If the z-axis movement is not clear, it would fallback to its 2D version (ex: PCCW → D2U). Otherwise if the performed gesture lacks of x- or y-axis movement, it would most likely recognized as N2F or F2N. HMM method doesn't work well on circulation gestures such as CW and CCW because of the state transitional nature of the model. HMM may recognize only one of the 4 possible starting point of CW and CCW gestures.

EVALUATION

We conducted an usability test to find out if the performance is acceptable when the gesture is put in practical use. 8 participants (5 male, 3 female) were recruited to perform 2 tasks. One task is to manipulate Google Map application. Starting from the entrance of our campus, the user are asked to move the view to our department using linear and rotation movement gestures, CW, and CCW. The other task is using the rotation gestures (CCW, CW, RCW, RCCW, PCW and PCCW) to find out the colors of the 6 surfaces of a 3D cube.

Before the test, we demonstrated all the 12 gestures to the participant. A practice time of at most 5 minutes were given to the participant. The test starts as soon as the participants were familiar and comfortable with the gestures. After the test, a short interview collects the qualitative feedback of the participant.

4 questions are asked in the short interview: 1) what do you think about the recognition time and the accuracy? 2) Is the gesture design straightforward? 3) What do you think about the trigger method? 4) What gestures are awkward to perform? What gestures can be performed smoothly? 5) Could you think of an application for the rotation gestures (i.e. RCW, RCCW, PCW, PCCW) ?

Result

5 of 8 participants agree that the accuracy of the gesture recognition is acceptable. However, 4 of the participants do not support nor against that our gesture design is straightforward. 3 of them pointed out that in the cube application, they cannot imagine their movements crossing behind the cube. It seems more intuitive for them to swipe the 3D cube, as the cube being placed somewhere ahead in a distance. 4 participants thought the response time of gesture recognition is not desirable. Two of them expected that the response should be synchronous to the screen response. For example, the map application should pan at the same time the user performed the gesture, not after the gesture was finished.

Two participants mentioned that the triggered hint really helped them performing recognizable gestures. "But the visual feedback should not be too eye-catching", said participant P3, "Maybe one little indicator is good enough. I do not like to be distracted from what we are doing." 3 out of 8 participants mentioned the trigger involves cognitive efforts, especially for F2N. However, another participant pointed out that F2N gesture is actually quite easy to perform after her getting used of this.

3 users prefer the linear gestures (L2R, etc.). Two of the participants thought CW and CCW are quite easy to perform, owing to its low error rate and that it is clutch-free. P1 reported that rotations involving back-and-forth (z-axis) movements lead to fatigue. P2 pointed out that she would rather perform only a part of the rotation movements; a semi-circle arc or a quarter-circle arc should be enough to express the rotation command.

When asked upon the application of 3D gestures, 5 users thought of 3D programs like 3D model viewers. Interactions between users and the 3D model before a 3D display screen can be used in indoor navigation, astronomy exploration, or 3D product showcase.

DISCUSSION

From the usability test we observed that if the body height much higher or lower to the average height of training data. Poses which participants are used to also plays a big role in recognition accuracy. If a particular pose have not been presented in the training data, it is most unlikely that the pose would be correctly identified.

Personalized model may help improve the recognition accuracy. We found that the personal gesture movements are consistent during different trials. Therefore, adaptive method may be applied for gaining a higher accuracy while not requiring a bootstrap training phase.

Though ultrasonic sensors are immune to lighting condition and noisy environment, there are some limitations. First, the directivity of ultrasonic sensor is not as high as the proximity sensor; the beam cone would diverge after a certain distance from the source and leads to some false positive recognitions. In addition, the ultrasonic rangefinders may interfere other ultrasonic rangefinders in the same space if they are not properly synchronized.

CONCLUSION AND FUTURE WORK

We have presented SoundSense, a 3D gesture recognition system on mobile devices. We completed a 16-person design study to understand the behavior of users for such gestures. A functional prototype is implemented on tablet supporting real-time recognition and operations. Gesture delimiter is introduced for solving the clutching problem. We compared two recognition methods: SVM and HMM, and concluded that SVM with DTAK method leads to the best result. Qualitative study shows the usability of the system on both map and 3D model applications.

While only uni-manual gestures are discussed in the paper, it is possible to extend our work to bi-manual gestures. In addition, the distance information may be further extended as a continuous value controller, which may be desirable for applications like volume control.

REFERENCES

1. Arentz, W.A. and Bandara, U. Near ultrasonic directional data transfer for modern smartphones. *Proceedings of the 13th international conference on Ubiquitous computing*, ACM (2011), 481–482.

2. Benko, H. Beyond flat surface computing: challenges of depth-aware and curved interfaces. *Proceedings of the 17th ACM international conference on Multimedia*, (2009).

3. Bihler, P., Imhoff, P., and Cremers, A.B. SmartGuide – A Smartphone Museum Guide with Ultrasound Control. *Procedia Computer Science 5*, (2011), 586-592.

4. Borriello, G., Liu, A., Offer, T., Palistrant, C., and Sharp, R. WALRUS: wireless acoustic location with room-level resolution using ultrasound. *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, ACM (2005), 191–203.

5. Chang, C.-chung and Lin, C.-jen. LIBSVM : A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST) 2*, 3 (2012), 1-39.

6. Gupta, S., Morris, D., and Patel, S. SoundWave: Using the Doppler Effect to Sense Gestures. *Proceedings of the 30th international conference on Human factors in computing systems*, (2012), 4-7.

7. Hsu, C.-W. and Lin, C.-J. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on 13*, 2 (2002), 415-425.

8. Kim, J., He, J., and Lyons, K. The gesture watch: A wireless contact-free gesture based wrist interface. *Wearable Computers, 2007 11th IEEE International Symposium on*, (2007).

9. Kreczmer, B. Gestures recognition by using ultrasonic range-finders. *Methods and Models in Automation and Robotics (MMAR), 2011 16th International Conference on*, (2011), 363-368.

10. Lyons, K., Brashear, H., Westeyn, T., Kim, J., and Starner, T. Gart: The gesture and activity recognition toolkit. *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments*, (2007), 718–727.

11. Malacria, S. and Lecolinet, E. Clutch-free panning and integrated pan-zoom control on touch-sensitive surfaces: the cyclostar approach. *Proceedings of the 28th international conference on Human factors in computing systems*, (2010), 2615-2624.

12. Mariappan, M.B., Guo, X., and Prabhakaran, B. PicoLife: A Computer Vision-based Gesture Recognition and 3D Gaming System for Android Mobile Devices. *Multimedia (ISM), 2011 IEEE International Symposium on*, (2011), 19-26.

13. Shimodaira, H., Noma, K.I., Nakai, M., and Sagayama, S. Dynamic Time-Alignment Kernel in Support Vector Machine. *Advances in Neural Information Processing Systems 14 2*, 1 (2006), 921-928.

14. Vogel, D. and Balakrishnan, R. Distant freehand pointing and clicking on very large, high resolution displays. *Proceedings of the 18th annual ACM symposium on User interface software and technology*, (2005), 33.

15. Wilson, A.D. Robust Computer Vision-Based Detection of Pinching for One and Two-Handed Gesture Input. *Proceedings of the 19th annual ACM symposium on User interface software and technology*, (2006), 255-258.