Dolphin: Ultrasonic-based Gesture Recognition on Smartphone Platform

Yang Qifan, Tang Hao, Zhao Xuebing, Li Yin
College of Software Engineering
Southeast University
Nanjing, China
{yqf3139, whiteT, xbzhao, yinli}@seu.edu.cn

Zhang Sanfeng*
Key Laboratory of Computer Network and Information Integration (Southeast University)，Ministry of Education
Southeast University
Nanjing, China
sfzhang@seu.edu.cn

***Abstract*—User experience of smart mobile devices can be improved in numerous scenarios with the assist of in-air gesture recognition. Most existing methods proposed by industry and academia are based on special sensors. On the contrary, a special sensor-independent in-air gesture recognition method named Dolphin is proposed in this paper which can be applied to off-the-shelf smart devices directly. The only sensors Dolphin needs are the loudspeaker and microphone embedded in the device. Dolphin emits a continuous 21 KHz tone by the loudspeaker and receive the gesture-reflecting ultrasonic wave by the microphone. The gesture performed is encoded into the reflected ultrasonic in the form of Doppler shift. By combining manual recognition and machine leaning methods, Dolphin extracts features from Doppler shift and recognizes a rich set of pre-defined gestures with high accuracy in real time. Parameter selection strategy and gesture recognition under several scenarios are discussed and evaluated in detail. Dolphin can be adapted to multiple devices and users by training using machine learning methods.**

Keywords—In-air gesture recognition ; Ultrasonic; Doppler; Interaction Technique;

I. INTRODUCTION

Human–Computer Interaction (**HCI**) based on gesture recognition is increasingly important for smart mobile devices such as phones or tablets. Compared to HCI based on touch screen [1], voice, electromyography sensors [2] and brainwave sensors, in-air gesture recognition show special advantages in several scenarios. It does not need direct contact with the device and is easy to learn. We have seen lots of efforts been put to in-air gesture recognition in recent years.

The novel gesture recognition techniques include those based on handheld controllers, data gloves [3], video or infrared image signals [4, 5] and ultrasonic. The approach based on ultrasonic is capable of performing low computational gesture recognition without special wearable equipment. It is also not disturbed by ordinary noise or light. This approach provides a wider operating range and angle, thus able to be populated on portable devices.

Gupta et al. [6] have undertaken preliminary work to explore the possibility of gesture recognition using ultrasonic, but the study on gesture recognition using existing smart mobile devices without extra sensors has not been reported.

This paper presents a gesture recognition technique using the loudspeaker and microphone embedded in smart devices as ultrasonic I/O devices. The technique utilizes the Doppler shift of the ultrasonic reflected by moving human body. The system samples the ultrasonic continuously while a gesture is performing. It extracts a time-varying sequence which is rich in unique features of every gesture. Then we classify gestures by combining simple pattern matching and supervised machine learning methods. Smart devices can achieve an average classification accuracy of 94% over a rich set of 24 pre-defined gestures. Continuous gestures are also defined. The frequency of waving can be calculated and used for complex control operations.

We validate the technique on Android devices and develop an Android system plug-in to provide gesture recognition services. Users can define and train gestures on their own to suit their needs. They can map the gestures to operations in third-party applications. For example, the swiping or continuous slapping gesture can be used to scroll web pages, flip e-books, pause movies or control interactive games. We also design two real-time games to show low latency and high accuracy of Dolphin.

The paper is organized as follows: section II describes related work in ultrasonic gesture recognition and introduces the originality of our work; section III introduces Dolphin system framework including recognition principles, recognition process and hardware settings; section IV describes feature extraction, gesture recognition and gesture definition in details; in section V, we evaluate the accuracy of manual classification under different environment settings. Different feature extraction strategies are put into comparison. Different classification methods are evaluated to find the best one both in accuracy and consumption; section VI concludes our contributions and proposes future plan.

II. RELATED WORK

We are not the first to use ultrasonic Doppler shift to monitor and classify human activity. Gupta et al. [6] track the Doppler shift caused by human activity on commercial laptops. With five gestures predefined, they achieve 94% gesture classification accuracy on multiple laptops in different environment sessions. They propose that it is possible to

*Corresponding author
This work is supported by the National Natural science Foundation of China under Grant No. 61300200 and 61472080

978-1-4799-7981-3/14 $31.00 © 2014 IEEE
DOI 10.1109/CSE.2014.273

measure the velocity, sectional area and direction of the moving target. Most techniques rely on peripherals. Kalgaonkar et al. [7] use low-cost ultrasonic transducers to recognize gestures in 3D space. By placing three receivers in a triangle pattern and a transmitter in the center, they are able to achieve accuracy of 88.42% on a set of eight gestures performed. However, their methods cannot be directly applied to smart mobile devices due to the limit of platform and special sensors.

Ultrasonic is also used to monitor the attention and gait of individual. Tarzia et al. [8] create an ultrasonic environment. Then they analyze the audio recorded by a laptop to measure user presence and attention. Kalgaonkar et al. [9] identify speech activity by emitting an ultrasonic beam on the talker's face. They also try to recognize individual identity by analyzing the Doppler shift while people are passing by the transmitter and receiver [10]. Uegami et al. [11] using Doppler sensors to develop an embedded device which can detect trip and fall of elderly person. Dura-Bernal [12] try to classify 7 activities in daily life, such as clapping hands and riding a bike. Watanabe et al. [13] focus on applying ultrasonic on wearable computing. They place several ultrasonic hotspots emitting different frequency in the context. The user can build a life log and recognize the context by wearing a set of peripherals including ultrasonic transmitter and recorders. The methods they put forward have great reference value for us.

Compare with the work of Gupta et al., we introduce a gesture recognition system which combines manual and machine classification. The system can recognize more gestures and achieves higher accuracy on the Android platform. Gesture recognition on mobile devices are troubled from lack of computing resources and the complexity of device structure. However, the method based on training can be adapted to different devices and different user preferences.

III. SYSTEM FRAMEWORK

Considering the variation in size and structure of smart mobile devices, it is challenging to recognize gestures using only loudspeaker and microphone. We track the tiny Doppler shift reflected by individual to decode the activity of human body. In this section, we discuss the hardware settings and recognition principles behind Dolphin.

A. Recognition Principle

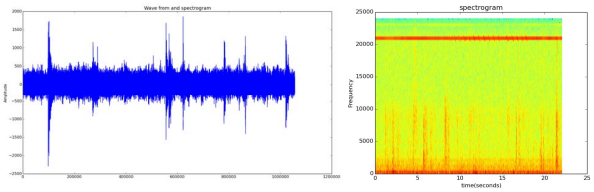
To recognize gestures based on ultrasonic Doppler shift, an ultrasonic tone should be emitted and then the reflected sound wave should be sampled properly.

Ultrasonic is the sound wave which has the frequency over 20 KHz. It has good directivity, nice penetration ability and focusable energy. The Doppler shift is the change in frequency of a wave for an observer moving relative to its source. Compared to the emitted frequency, the received frequency is higher during the approach, identical at the instant of passing by, and lower during the recession.

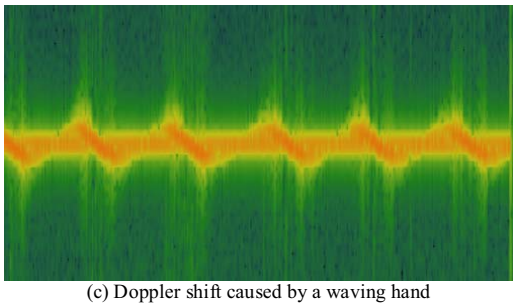
According to the Doppler formula, assuming the device as both the ultrasonic source and receiver, and the human body as the reflecting surface, we have:

$$f_r = f_t \cdot \frac{v_s + v_p}{v_s - v_p}, \tag{1}$$

Where f_r is the frequency of received ultrasonic, f_t is the frequency of transmitted ultrasonic, v_t is the speed of sound in air, and v_p is the speed of individual gesture. We can deduce that f_r is proximately 21123.79 Hz when v_s = 340.29 m/s, v_p = 1m/s and f_t = 21000 Hz. The human body is now approaching the devices, and the Doppler shift is positive.



(a) time-domain signal (b) frequency domain signal



(c) Doppler shift caused by a waving hand

Fig. 1. Doppler Effect Visualization

Fig.1(a) shows the original time-domain signal. The abscissa is the time and the ordinate is the frequency. Fig.1 (b) (c) shows the spectrum at a particularly sample time. The abscissa is the frequency, and the ordinate is the intensity. The shading of the color represents intensity. The wave-like Doppler shift is caused by a regularly waving hand.

The direction, velocity and the size of the moving target can be deduced using formula (1). Gestures can be recognized by analyzing the continuous sequence of Doppler shift.

Users can move their hands passing the front or in the plane of the device to perform a gesture. Since the gesture is not limited to a specific angle or a particularly space, it is more practical in daily use. The gesture will not be tracked if the distance between the device and user is more than one meter. Note that the distance threshold can prevent interference from other people.

B. Recognition Process

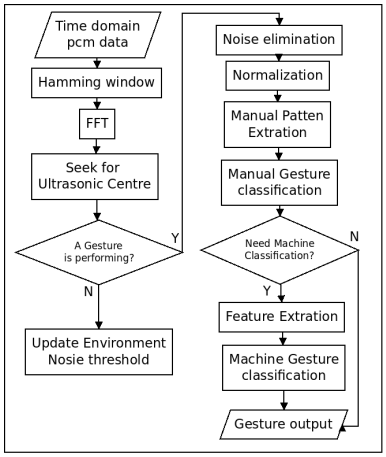


Fig. 2. The process of Dolphin gesture recognition

First, we acquire time-domain signals from microphone. An ultrasonic data vector is acquired using Fast Fourier Transform (**FFT**) conversion and the ultrasonic center can be located; then the vector is categorized into gesture signal or noise judging by recent spectrum data. If an environment noise vector is received, the environment noise threshold will be updated. If the gesture signal is received, preprocessing of the signal will be performed. Features of the gesture will be extracted for further analysis; finally we combine manual classification and machine classification to get optimized gesture result.

C. Hardware and Data Acquisition

Fig.3 shows an original smart phone (MI One). The loudspeaker is located beneath the phone. The microphone is located on the left underside while its handset is at the top front of the device. In general, the position of the earphone and the microphone is fixed for most devices. However, the loudspeaker may appear in front, back or bottom of the device. To make sure the ultrasonic interfere with the human body, we choose the loudspeaker or the earphone to emit ultrasonic wisely.

We use algorithm to generate the 16 bit-wide audio of the pure frequency of 21 KHz on MI One. The ultrasonic tone is then played continually. We sample at a rate of 48 KHz. The data vector we acquired is 48 KHz 16 bit-wide time-domain signal.

The ultrasonic emitted by the speaker will create an ultrasonic environment around the device which we called Dolphin Bubble. The gesture performed in this bubble will be captured by the device, even a slightly movement of fingertips.

Fig. 3. Dolphin testing platform MI One

More compatibility work should be done because the microphone we use is not specific for sampling ultrasonic. The sampling performance also vary between devices. The original signal is hamming windowed and then a 4096 point FFT transform is preformed to acquire a 2049 point spectrum vector.

IV. Implementation Details

In this section, we focus on several important steps in gesture recognition including feature extraction, gesture classification and gesture definition.

A. Perprocessing

The goal of the preprocessing is to transform the original Doppler shift data into a normalized, noise-eliminated, width-fixed ultrasonic vector.

1) *Obtaining ultrasound data vector*

First, the original data vector O_t at the sample time *t*, is produced by the FFT transform. The center frequency of the emitted tone can be found according to the intensity of each frequency bin. As Fig.4 (a) shows, the peek emerges around 1780. First we save 60 points around the peak as the initial ultrasonic vector V_t ; then V_t is preprocessed as shown in Fig.4(b); finally we judge whether a gesture is performed, otherwise the environment noise threshold is updated using V_t .

2) *Noise elimination*

In order to eliminate noise not related to gestures, we need to maintain an environment noise threshold vector N . The threshold vector is initiated during the preparation process and is updated during the gesture gap:

$$N_t = N_{t-1} \cdot (1-\alpha) + E_t \cdot \alpha \qquad E_t, N_t \in R^{60\times 1} \tag{2}$$

$$I_t' = I_t - N_t \tag{3}$$

Where N_t is the updated noise threshold vector at time t.

E_t is the environment noise vector at time t. α is 0.1.

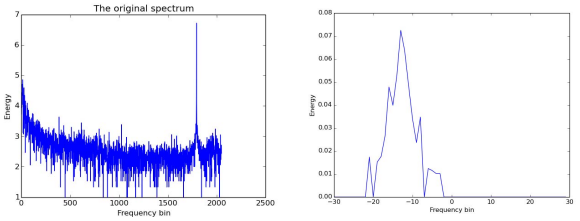
I_t is the original data vector and I_t' is noise-eliminated.

3) *Normalization*

Normalization need to be performed on I_t' to avoid the uncertainty of ultrasonic intensity.

$$S_t = \sum_{i=L_t}^{H_t} I_t'(i), V_t(i) = \frac{I_t'(i)}{S_t} \tag{4}$$

Where H_t and L_t are the shift upper bound and lower bound respectively, S_t is the sum of frequency intensity from the shift lower bound to upper bound, V_t is the normalized ultrasonic data vector.



（a）Original spectrum （b）Shift data after preprocessing

Fig. 4. Data sampled at time *t*

B. Feature Extraction

1) Feature extration for manual classification

Manual classification is based on the sequence of the gesture direction transformed from sequence V_t . First, a weighted frequency shift value F_t should be calculated for every sample point:

$$F_t = \sum_{i=L_t}^{H_t} i \cdot V_t(i) \tag{5}$$

Where F_t represents the changes the target user have on the frequency center at time *t*, V_t is the reprocessed ultrasonic shift data vector.

First, we pick out the F_t sequence of a complete gesture ordered by time; then compress it into a shift sequence M , which consists only of discontinuous -1 and 1. When the shift is positive, which indicates the observer is moving relative to the device, we call the current state as 1. State is -1 when the shift is negative. For example, M will be [1,-1] if a push-pull gesture is performed.

2) Feature extration for machine classification

Machine classification is based on the proposed ultrasonic data vector I . First, I of a complete gesture are arranged in order of time to form a sequence of vectors $\{\mathrm{I}_{t_0}, \mathrm{I}_{t_1}, ...\}$. The sequence is then interpolated to fit into a fixed length in order to eliminate the variations in speed. For instance, 30 vectors, each consists of 60 frequency bin, form a feature vector of total 1800 points:

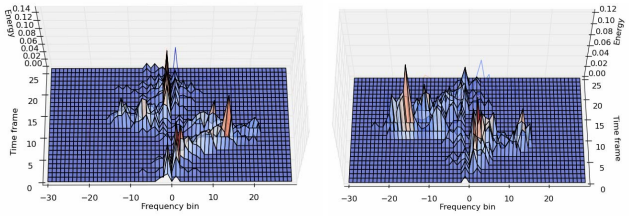
$$V = [\mathrm{V}_0{}^T, \mathrm{V}_1{}^T, ..., \mathrm{V}_{29}{}^T]^T, V \in R^{1800\times 1}, V_n \in R^{60\times 1} \tag{5}$$

The feature vector V is then used for training classifier and recognizing gestures.

C. Gesture Classification

In order to classify more gestures with high accuracy, we introduce a strategy combing manual and machine classification. Because the Doppler shift only indicates relative distance changes between human body and device, several gestures may look the same in this perspective. For example, swiping to left, swiping to right and push-pull gesture has the same shift sequence [1,-1] for they all approach and then recess. As a result, they are all classified as push-pull gesture. To classify these gestures in finer granularity, we have to analyze more detailed time–varying features. Fig.5 shows the 3D spectrum of swiping to right and swiping to left. We found that the characteristics of the gestures are greatly related to the structure of the device and the preference of the user, which is difficult for manual classifier to handle. For MI One, swiping right makes more positive shifts than negative shifts.

As gesture categories and samples counts increase, the machine classification accuracy decreases and classification complexity rises. As a result, it is impossible to classify a set of 24 gesture using one classifier. Our solution is as follows: first, we categorize the gestures into gesture groups using manual gesture recognition; then we use machine classifier to classify gestures into finer granulated gesture.



(a) (b)

Fig. 5. Spectral characteristics of swiping to right (a) and left (b) on MI 1

TABLE I. ALGORITHM 1

Algorithm.1 Gesture Classification
Input: reprocessed ultrasonic vector
Output: gesture index
vector ← GetUltrasonicVector()
patten ← GetManualPatten(vector)
index ← MatchManualPatten(patten)
if NeedMachineClassification(index)
feature←GetFeatureVector(vector)
index←MachineClassification(index, feature)
endif

Algorithm.1 shows a combination of manual and machine classification.

D. Manual Gesture Classification

TABLE II. DEFINITION OF MANUAL CLASSIFIED GESTURE

Gesture	Description
|← (N)	Push your hand towards the device
|→ (F)	Pull your hand from the device
|←→ (NF)	Push your hand towards the device, then pull back
|→← (NF)	Pull your hand from the device, then push forward
|↔← (NFN)	Push your hand towards the device, then pull back, push forward eventually
|→↔ (FNF)	Pull your hands from the device, then push forward, pull back eventually
|↔↔ (NFNF)	Do |←→ twice
|→↔← (FNFN)	Do |→← twice

↕（CR）	Moving hands in opposite direction simultaneously
↕↕（CT）	Waving your hand continually

We define a set of 10 gestures (as shown in Table Ⅱ) for manual classification. Each gesture corresponds to a sequence coded by -1 or 1. After each gesture is performed, we match the gesture with predefined gesture sequences and obtain the gesture index. For gestures that can be further classified, we generate the feature vector for the target gesture and obtain finer granulated result using machine classifier.

E. Machine Gesture Classification

For the gesture instance classified by manual classifier, we take further step to distinguish gestures sharing the same shift sequence. First we collect a certain number of samples of each gesture from the target device. The optimal number of sample is 20; then the samples of the same gesture group are used to train the classifier.

The classifiers we evaluated are as follows: *Native Bayes; IBK (K-nearest neighbor classifier); Bayes Net; Random Tree; Liblinear (Large Linear classifier); SVM; AN (Artificial network or)*.

The classifier we choose is the Liblinear classifier with both high accuracy and low energy consumption for our settings; finally, we use the pre-trained classifier to classify each gesture group and label the most likely gesture as result.

TABLE III. GESTURE GROUP OF NF

Gesture	Description
|←→（NF）	Push your hand towards the device, then pull back
←（SWIPELL）	Swipe your hand from right to left edge of the device with the device placed in landscape mode
→（SWIPERL）	Swipe your hand from left to right edge of the device with the device placed in landscape mode
←（SWIPELP）	Swipe your hand from right to left edge of the device with the device placed in portrait mode
→（SWIPERL）	Swipe your hand from left to right edge of the device with the device placed in portrait mode

TABLE IV. GESTURE GROUP OF FN

Gesture	Description
|→←（FN）	Pull back your hand from the device, then push forward
↙↗（SWINGLL）	Swipe your hand from middle to left edge of the device with the device placed and then swipe back in landscape mode
↘↖（SWINGRL）	Swipe your hand from middle to right edge of the device with the device placed and then swipe back in landscape mode
↙↗（SWINGLP）	Swipe your hand from middle to left edge of the device with the device placed and then swipe back in portrait mode
↘↖（SWINGRL）	Swipe your hand from middle to right edge of the device with the device placed and then swipe back in portrait mode

TABLE V. GESTURE GROUP OF NFNF

Gesture	Description
|↔↔（NFNF）	Do |←→ twice
←→（SBACKLL）	Swipe your hand from right to left edge of the device and then swipe back with the device placed in landscape mode
→←（SBACKRL）	Swipe your hand from left to right edge of the device with the device placed and then swipe back in landscape mode
←→（SBACKLP）	Swipe your hand from right to left edge of the device and then swipe back with the device placed in portrait mode
→←（SBACKRP）	Swipe your hand from left to right edge of the device and then swipe back with the device placed in portrait mode

TABLE VI. GESTURE GROUP OF CR

Gesture	Description
↕（CR）	Move hands in opposite direction simultaneously
↑↓（CRL）	Push your left hand towards and pull your right hand back from the device simultaneously
↓↑（CRR）	Push your right hand towards and pull your left hand back from the device simultaneously

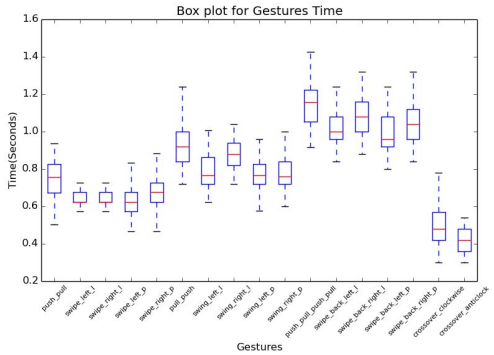


Fig. 6. Box plot of the gesture duration

The gestures is clearly divided into 4 groups, as can be learned from Fig.6. However, the distribution of the gesture duration is close to each other in the same group. For example, ← and → have the same distribution in duration and the same shift sequence. As a result, it is not enough to classify all the gestures in a gesture group depending only on the shift sequence and the duration. By using machine leaning methods to study the variation of ultrasonic shift data, we can classify gestures of a gesture group with high accuracy.

V. EVALUATION

We evaluate the accuracy of manual and machine recognition on different devices. We select the optimal features and classifier for machine classification. We discuss factors which affect the accuracy of gesture recognition in detail. In order to show the performance of Dolphin in daily use, we develop a system plugin for Android. Using Dolphin, the user can map their in-air gestures into third-party applications. We also develop two real-time control games to show practicality of Dolphin.

A. *Evaluate the accuracy of manual classification*

We evaluate the accuracy of manual classification under the circumstances of different devices, environments, users and postures. Three Android devices are used including two phones and one tablet (MI One, Samsung S3 and Nexus 7 2013). The environment we choose are labeled as quiet, outdoor and noisy. We have three users participated in the evaluation process with the device on the table or in hand.

The environment labeled quiet is an indoor setting with no noise in the high frequency region and a small amount of low frequency noise. Dolphin is completely free from noise in this session. The environment labeled outdoor is a setting with some noise in the high frequency region. The environment labeled noisy is a setting with noise in all frequency region. Dolphin is able to sense gestures by increasing the volume of ultrasonic.

Table Ⅶ shows manual classification accuracy in detail. If the device is held in hand by the user, the crossover gesture cannot be performed.

TABLE VII. STATISTICS OF CORRECTLY RECOGNIZED GESTURES FOR MANUAL GESTURE RECOGNITION UNDER 6 ENVIRONMENT SETTINGS

Environment	Position	Cluster 1	Cluster 2	Cluster 3	Cluster4	Others	Average
Normal	Table	100%	100%	100%	100%	93.75%	98.81%
	Hand	100%	98.33%	96.66%	-	91.66%	96.93%
Outdoor	Table	100%	100%	100%	100%	95.83%	99.21%
	Hand	98.33%	96.66%	98.33%	-	87.5%	95.61%
Extreme Noisy	Table	78.33%	86.66%	85%	91.66%	81.25%	83.73%
	Hand	71.66%	83.33%	86.66%	-	77.08%	79.82%

We find that manual gesture classification is not disturbed both indoor and outdoor. The accuracy is about 95%. So the manual gesture recognition may works well in the cafe. But for several extreme noisy environment, such as in the canteen or on the subway, the recognition for short-duration gesture may be disturbed. It is also easy to be interfered while the user is walking and performing gesture at the same time.

B. *Selection of features*

A gesture dataset has been set up to evaluate the effect of three parameters on gesture recognition accuracy. The dataset contains a total of 17 kinds of gestures which categorized into four groups. We collect 110 samples for each gesture.

Trained instance number: Number of trained samples for each gesture, value interval is [5, 30], the n samples are extracted as the training set from the 110 samples and the remaining samples as the test set.

Single width: Half the number of the data points, value interval is [10, 30].

Time counter: Count of time frames, value interval is [10, 30], the sequence is interpolated to fit into a uniform length in order to eliminate the variations in speed of the gesture.

1) *Selection of the trained samples*

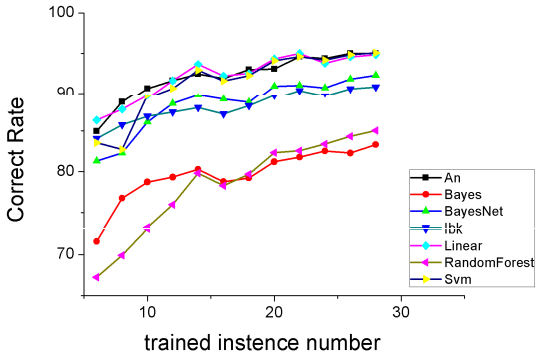


Fig. 7. Accuracy of classifiers changes as trained sample number increases

Fig.7 shows the trend of the accuracy as the trained samples number increases. As can be seen, although the accuracy of different classifiers are not the same, they share the same trend that the rate of change continues to reduce. The trend indicates that it is possible to categorize gestures accurately by learning a small amount of samples (less than 20 samples).

AN, Liblinear and SVM has the highest accuracy while the accuracy of Bayes Net and IBK is approaching 90%.The rate of increasing curve slows down as the sample number is over 15. The three top classifier share the accuracy of 93% when the training sample number is 20. As the trained sample number decreases, the cost of training decreases, which is easy to implement incremental learning on smart mobile devices. Power consumption is also reduced. Here we use 20 samples as the optimal number of training samples.

2) *Selection of the doppler shift data width*

More Doppler shift data can be included as the length of the data vector increases. However, the possibility of the large shift is small. The training cost is increasing as more data points are included, as shown in Fig.8. This can even produce side effects for some classifiers. Here we use 20 as the optimal width in order to achieve a balance between accuracy and power consumption.

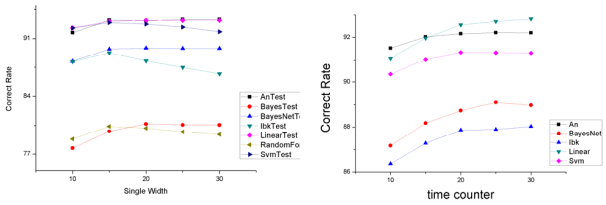


Fig. 8. Accuracy of classifiers as single width increases

Fig. 9. Accuracy of classifiers as time counter increases

3) *Selection of duration of the gesture samples*

Fig.9 shows that accuracy changes as time counter increases. More details will be preserved in the feature vector as time counter increases. The longer the time frame length, the greater the complexity of the training. We choose 20 as the optimal time frame length.

C. *Selection of classifiers*

We choose classifier according to the average accuracy, training time and classification time using features extracted with different parameter of time counter, single width and trained instances.

1) Accuracy

Fig.10 shows the average accuracy for each classifier. Liblinear and AN share the highest classification accuracy.

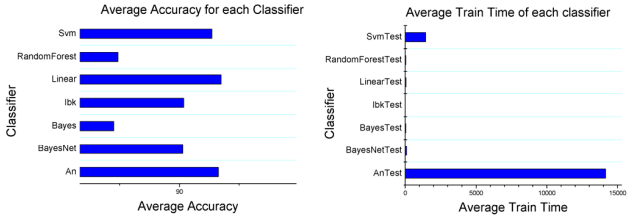


Fig. 10. The average accuracy for each classifier

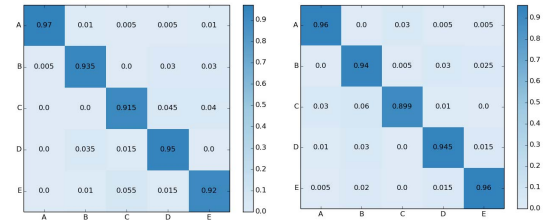
Fig. 11. The average classification time of each classifier

2) Classification time and training time

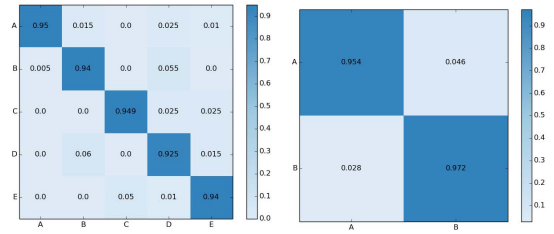
Fig.12 shows the average classification time and training time. Liblinear and AN have low classification time, but the training complexity of AN is high.

Finally, we choose Liblinear as our machine classifier.

3) Confusion matrix of Liblinear using optimal features



(a) NF gesture group (b) FN gesture group



(a) NFNF gesture group (b) CR gesture group

Fig. 12. Confusion matrixes for 4 gesture groups

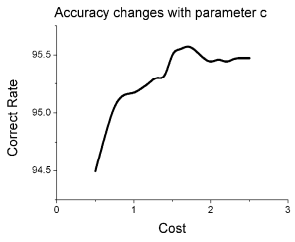


Fig. 13. Accuracy changes as parameter c changes

4) Liblinear parameter tuning

By adjusting the cost parameter in Liblinear, the accuracy can be improved slightly as Fig.13 shows. Here we choose 2 as the default cost value.

D. *Optimization using gravity sensor*

The device can be placed in lying or upright mode in actual use, thus the same gesture may vary in duration or shift sequence. For example, the swiping gesture is differ from each other between the lying mode and the upright mode. If user changes the device position, the accuracy may drop. However, training more samples of the lying mode and the upright mode separately will increase the training complexity.

As shown in Table.Ⅷ, we find the accuracy to be 90% if we use the existing gravity sensor to assist the classification process. The accuracy drop to 81% using only one classifier with more samples provided. The accuracy drop to 74% if we use only 20 samples to train the classifier.

TABLE VIII. ACCURACY OF DIFFERENT CLASSIFICATION STRATEGIES

Using Gravity Sensor	Trained instance number per cluster	Correct Rate
Yes	20	89.26%
No	60	81.17%
No	20	73.27%

E. *False-positive rate with no gestures performed*

Because of movement of user or noise, gesture may be recognized while no gesture is performed actually. We count and analyze the false-positive gestures with people present or not.

Table Ⅸ shows the count of false positive gestures recognized under several environment settings in the time internal of half an hour.

Under the unmanned environment, the false-positive rate is nearly zero. Because of the noise-elimination process, it is difficult for noise signals to satisfy the standard of a gesture.

TABLE IX. FALSE-POSITIVE COUNTS UNDER SEVERAL ENVIRONMENT

Environment	Manned	Gesture	Count
Study	no	-	0
	yes	|←	14
		|→	6
Bedroom	no	-	0
	yes	|←	5
		|→	2
Kitchen	no	|→	1
	yes	|→	34
		|←	29
		→t	4
		↕	4

The false positive rate is directly proportional to the intensity and distance of human activity. However, the gestures recognized are mainly single push and pull. The possibility of a false-positive judgment on a complex gesture is small. The

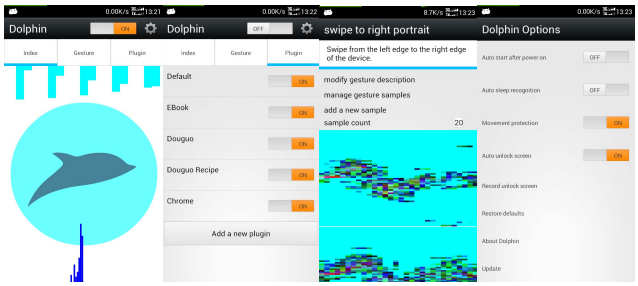
false positive rate can be reduced by simply dropping the simple gestures or perform a startup gesture.

F. *The practical applications of Dolphin*

The predefined 24 gestures can be used to bring user a rich HCI experience with smart mobile devices. For example, the swiping or continuous slapping can be used to scroll webpages, flip e-books, pause movies or control interactive games. The in-air gesture can work seamlessly with the screen and be complementary to touch control.

More gestures can also be defined and added to the system by training more samples of gestures.

In order to fully demonstrate the performance of Dolphin, we designed and implemented an Android system plugin and two games. Reading e-books, webpages and controlling games are both responsive and accurate from a practical point of view.

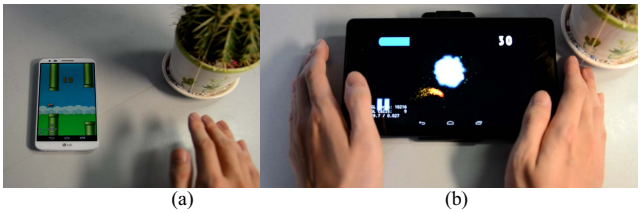


(a) (b) (c) (d)

Fig. 14. Dolphin system plug-in interface

Fig.14 shows the Dolphin system plugin which provides the features of ultrasonic visualization, gesture sample management and third-party application customization. .

Fig.15 shows the interactive games Dolphin supports. User can flap their hands to make the bird to fly across the obstacles. User can control the attraction and repulsion of particles to absorb energy particles and avoid destructive particles.



(a) (b)

Fig. 15. Games using Dolphin gesture recognition

VI. Conclusion and Future Work

This paper presents a system providing extra-sensor-independent in-air gesture recognition services to off-the-shelf smart devices. The system provides gesture recognition service of 10 groups and 24 gestures. It can achieve the average accuracy of 93% under two normal environment settings. There is still more work to be done in the future: To improve the accuracy and fault tolerance performance by elaborating gesture definitions; To analyze and optimize the performance in those scenarios with complicated background noises, such as waiting room, buses on the road, high-speed rail and cafes; To analyze and optimize the energy consumption which is not negligible on mobile platform.

VII. References

[1] P.H Dietz and D.L. Leigh, “Diamondtouch: A multi-user touch technology,” ACM UIST, pp. 219–226, 2001.

[2] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, “A framework for hand gesture recognition based on accelerometer and EMG sensors,” IEEE Transactions on Systems, Man, and Cybernetics Part A, vol. 41, no. 6, pp. 1064–1076, 2011.

[3] P. Kumar, J. Verma and S. Prased, “Hand Data Glove: A Wearable Real-Time Device for Human Computer Interaction,” J. International Journal of Advanced Science and Technology, vol. 43, no. 2, June, pp. 15-26, 2012.

[4] B. Kellogg, V. Tallat, S. Gollakota. “Bringing Gesture Recognition to All Devices,” In Proc. of Usenix NSDI’14, Seattle, USA, 2014.

[5] Q. Pu, S. Gupta, S. Gollakota, and S. Patel. “Whole-home gesture recognition using wireless signals,” In Proc. of MobiCom ’13.

[6] Gupta, S., Morris, D., Patel, S., and Tan, D. “SoundWave: Using the Doppler Effect to Sense Gestures,” In Proc. Of the SIGCHI Conference on Human Factors in Computing Systems 2012.

[7] Kalgaonkar, K. and Raj, B. “One-handed gesture recognition using ultrasonic Doppler sonar,” In Proc. Of IEEE Acoustics, Speech and Signal Processing 2009.

[8] Tarzia, S., Dick, R. Dinda, P. and Memik, G. “Sonar-based Measurement of User Presence and Attention”. In Proc. Of Ubicomp 2009.

[9] K. Kalgaonkar, Rongquiang Hu, and B. Raj, “Ultra-sonic doppler sensor for voice activity detection,” Signal Processing Letters, IEEE, vol. 14, no. 10, pp. 754–757, Oct. 2007.

[10] K. Kalgaonkar, R. Bhiksha, “Acoustic Doppler Sonar for gait recognition,” Mitsubishi Electric Research Laboratories, Inc., 2007.

[11] Uegami, Masaru, Takeshi Iwamoto, and Michito Matsumoto. "A study of detection of trip and fall using Doppler sensor on embedded computer." Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on. IEEE, 2012.

[12] Dura-Bernal, Salvador, et al. "Human action categorization using ultrasound micro-Doppler signatures." Human Behavior Understanding. Springer Berlin Heidelberg, 2011. 18-28.

[13] Watanabe, Hiroki, Tsutomu Terada, and Masahiko Tsukamoto. "Ultrasound-based movement sensing, gesture-, and context-recognition." Proceedings of the 17th annual international symposium on International symposium on wearable computers. ACM, 2013