Pervasive and Mobile Computing 5 (2009) 657–675

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

uWave: Accelerometer-based personalized gesture recognition and its applications

Jiayang Liu [a,*], Lin Zhong [a], Jehan Wickramasuriya [b], Venu Vasudevan [b]

[a] *Department of Electrical Computer Engineering, Rice University, Houston, TX 77005, United States*

[b] *Pervasive Platforms & Architectures Lab, Applications & Software Research Center, Motorola Labs, United States*

ARTICLE INFO

Article history:
Received 28 February 2009
Received in revised form 5 June 2009
Accepted 4 July 2009
Available online 18 July 2009

Keywords:
Gesture recognition
Acceleration
Dynamic time warping
Personalized gesture
User authentication

ABSTRACT

The proliferation of accelerometers on consumer electronics has brought an opportunity for interaction based on gestures. We present uWave, an efficient recognition algorithm for such interaction using a single three-axis accelerometer. uWave requires a single training sample for each gesture pattern and allows users to employ personalized gestures. We evaluate uWave using a large gesture library with over 4000 samples for eight gesture patterns collected from eight users over one month. uWave achieves 98.6% accuracy, competitive with statistical methods that require significantly more training samples. We also present applications of uWave in gesture-based user authentication and interaction with 3D mobile user interfaces. In particular, we report a series of user studies that evaluates the feasibility and usability of lightweight user authentication. Our evaluation shows both the strength and limitations of gesture-based user authentication.

1. Introduction

Gestures[1] have recently become attractive for spontaneous interaction with consumer electronics and mobile devices in the context of pervasive computing [1–3]. However, there are multiple technical challenges to gesture-based interaction. First, unlike many pattern recognition problems, e.g. speech recognition, gesture recognition lacks a standardized or widely accepted "vocabulary". It is often desirable and necessary for users to create their own gestures, or personalized gestures. With personalized gestures, it is difficult to collect a large set of training samples necessary for established statistical methods, e.g., Hidden Markov Model (HMM) [4–6]. Secondly, spontaneous interaction requires immediate engagement, i.e., the overhead of setting up the recognition instrumentation should be minimal. More importantly, the targeted platforms for personalized gesture recognition are usually highly constrained in cost and system resources, including battery, computing power, and interface hardware, e.g. buttons. As a result, computer vision [1,2] or "glove" [3] based solutions are unsuitable.

In this work, we present uWave to address these challenges and focus on gestures without regard to finger movement, such as sign languages. Our goal is to support efficient personalized gesture recognition on a wide range of devices, in particular, on resource-constrained systems. Unlike statistical methods [4], uWave only requires a single training sample to start; unlike computer vision-based methods [5], uWave only employs a three-axis accelerometer that has already appeared in numerous consumer electronics, e.g. Nintendo Wii remote, and mobile device, e.g. Apple iPhone. uWave matches the accelerometer readings for an unknown gesture with those for a vocabulary of known gestures, or *templates*, based on dynamic time warping (DTW) [6]. uWave is efficient and thus amenable to implementation on resource-constrained platforms. We

[*] Corresponding author.
E-mail address: jiayang@rice.edu (J. Liu).

[1] We use "gestures" to refer to free-space hand movements that physically manipulate the interaction device. Such movements include not only gestures as we commonly know; but also any physical manipulations like shaking and tapping of the device.

1574-1192/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.pmcj.2009.07.007

have implemented multiple prototypes of uWave on various platforms, including Smartphones, microcontroller, and the Nintendo Wii remote hardware [7]. Our measurement shows that uWave recognizes a gesture from an eight-gesture vocabulary in 2 ms on a modern laptop, 4 ms on a Pocket PC, and 300 ms on a 16-bit microcontroller, without any complicated optimization.

We evaluate uWave with a gesture vocabulary identified by a VTT research [4] for which we have collected a library of 4480 gestures for eight gesture patterns from eight participants over multiple weeks. The evaluation shows that uWave achieves accuracy of 98.6% and 93.5% with and without template adaptation, respectively, for user-dependent gesture recognition. The accuracy is the best for accelerometer-based user-dependent gesture recognition. Moreover, our evaluation data set is also the largest and most extensive in published studies, to the best of our knowledge.

We also evaluate the application of uWave in user authentication through a series of comprehensive user studies involving 25 participants over one month. Our user studies address two types of user authentication: non-critical authentication for a user to retrieve privacy-insensitive data and critical authentication for protection of privacy-sensitive data. For non-critical authentication, we demonstrate that uWave achieves average 98% accuracy with simple gesture selection constraints; a follow-up survey shows that the usability of uWave for non-critical authentication is comparable to the use of textual ID-based authentication. For critical authentication, we find 3% equal rate of false negatives, i.e. rejecting authentic users' gestures, and false positives, i.e. accepting attackers' gestures, or *equal error rate*, can be achieved without visual disclosure, meaning the attacker does not see the owner's password gesture performance. Visual disclosure increases the equal error rate to 10%. Therefore gesture-based authentication can be used only when strict security is either not necessary or can be achieved through combination of gesture-based authentication and traditional methods. Our evaluation highlights the need to conceal the gesture performance. Our analysis also shows the potential to achieve a lower equal error rate through recognizers that adapts to the users.

In summary, we make the following contributions.

- We present uWave, an efficient gesture recognition method based on a single accelerometer using dynamic time warping (DTW). uWave requires a single training sample per vocabulary gesture.
- We show that there are considerable variations in gestures collected over a long time and in gestures collected from multiple users; we highlight the importance of adaptive and user-dependent recognition.
- We report an extensive evaluation of uWave with over 4000 gesture samples of eight gesture patterns collected from eight users over multiple weeks for a predefined vocabulary of eight gesture patterns.
- We present two applications of uWave: gesture-based user authentication and gesture-based manipulation of three-dimensional user interfaces on mobile phones. The use of uWave in user authentication is extensively evaluated through a series of user studies.

The strength of uWave in user-dependent gesture recognition makes it ideal for personalized gesture-based interaction. With uWave, users can create simple personal gestures for frequent interaction. Its simplicity, efficiency, and minimal hardware requirement of a single accelerometer make uWave have the potential to enable personalized gesture-based interaction with a broad range of devices.

The rest of the paper is organized as follows. We discuss related work in Section 2 and then present the technical details of uWave in Section 3. We next describe a prototype implementation of uWave using the Wii remote in Section 4. We report an evaluation of uWave through a large database for a predefined gesture vocabulary of eight simple gestures in Section 5. We present the application of uWave to interaction with mobile phones and gesture-based user authentication in Sections 6 and 7 respectively. We discuss the limitations of uWave and acceleration-based gesture recognition in general in Section 8 and conclude in Section 9.

2. Related work

2.1. Gesture recognition

Gesture recognition has been extensively investigated [1,2]. The majority of the past work has focused on detecting the contour of hand movement. Computer vision techniques in different forms have been extensively explored in this direction [5]. As a recent example, the Wii remote has a "camera" (IR sensor) inside the remote and detects motion by tracking the relative movement of IR transmitters mounted on the display. It basically translates a "gesture" into "handwriting", lending itself to a rich set of handwriting recognition techniques. Vision-based methods, however, are fundamentally limited by their hardware requirements (i.e. cameras or transmitters) and high computation load. Similarly, "smart glove" based solutions [3,8,9] can recognize very fine gestures, e.g., the finger movement and conformation but require the user to wear a glove tagged with multiple sensors to capture finger and hand motions in fine granularity. As a result, they are unfit for spontaneous interaction due to the high overhead of engagement.

As ultra low-power low-cost accelerometers appear on consumer electronics and mobile devices, many have recently investigated gesture recognition based on the time series of acceleration, often with additional information from a gyroscope or compass. Signal processing and ad hoc recognition methods were explored in [10,11]. LiveMove Pro [12] from Ailive provides a gesture recognition library based on the accelerometer in the Wii remote. Unlike uWave, LiveMove Pro targets user-independent gesture recognition with a predefined gesture vocabulary and requires 5 to 10 training samples for each gesture. No systematic evaluation of the accuracy of LiveMove Pro is publicly available. HMM, investigated in [4,5,16,17], is the mainstream method for speech recognition. However, HMM-based methods require extensive training data to be effective. The authors of [13] realized this and attempted to address it by converting two samples into a large set of training data by adding Gaussian noise. While the authors showed improved accuracy, the effectiveness of this method is likely to be highly limited because it essentially assumes that variations in human gestures are Gaussian. In contrast, uWave requires as few as a single training sample for each gesture and delivers competitive accuracy. Another limitation of HMM-based methods is that they often require knowledge of the vocabulary in order to configure the models properly, e.g. the number of states in the model. Therefore, HMM-based methods may suffer when users are allowed to choose gestures freely, or for personalized gesture recognition. Moreover, as we will see in the evaluation section, the evaluation dataset and the test procedure used in [4,5,17] did not consider gesture variations over the time. Thus their results are likely to be overly optimistic.

Dynamic time warping (DTW) is the core of uWave. It was extensively investigated for speech recognition in the 1970s and early 1980s [6], in particular speaker-dependent speech recognition with a limited vocabulary. Later, HMM-based methods became the mainstream because they are more scalable toward a large vocabulary and can better benefit from a large set of training data. However, DTW is still very effective in coping with limited training data and a small vocabulary, which matches up well with personalized gesture-based interaction with consumer electronics and mobile devices. Wilson and Wilson applied DTW and HMM with XWand [14] to user-independent gesture recognition. The low accuracies, 72% for DTW and 90% for HMM with seven training samples, render them almost impractical. In contrast, uWave focuses on personalized and user-dependent gesture recognition, thus achieving much higher recognition accuracy. It is also important to note that the evaluation data set employed in this work is considerably more extensive than previously reported work, including [4,5,17]

It is important to note that some authors use "gesture" to refer to handwriting on a touch screen, instead of three-dimensional free-hand movement. Some of these works, e.g. "$1 recognizer" [15], were also based on template matching, similar to uWave. However, because they are based on matching the geometric specifications of two handwritings, it may not apply to matching time series of accelerometer readings, which are subject to temporal dynamics (how fast and forceful the hand moves), three-dimensional acceleration data due to movement of six degrees of freedom, and the confusion introduced by gravity.

2.2. User authentication

Most user authentication methods are based on either what properties the user has, e.g. fingerprint [16], face [17] and iris [18], or what he/she knows, e.g. password [19], or both, e.g. speaker verification [20] and handwritten signature recognition [21]. All these methods, however, require form factor modification or considerable computation and user engagement, unsuitable for operating small resource-constrained devices in a mobile manner. In contrast, accelerometer-based authentication allows free-space hand movement and does not require any form factor change to the device.

The work in [22,23] considers gesture as a behavioral biometrics that the user has and attempts to verify or recognize the user identity based on a fixed gesture performed by all participants, e.g. a simple arm swing in [22]. In contrast, we allow the user to create any physical manipulation of the device as the authenticating gesture. In other words, our authentication approach is based on both what the user "knows" and what properties the user has. As a result, our work investigates the human factors in gesture selection and the usability of customized gestures. The goal of [22] is similar to that of our critical authentication: to verify a claimed user identity. The authors showed about 4% equal error rate over long time but through adaptation with a large number of training samples [24], compared to 3% in our solution of critical authentication with a single training sample. Notably, the basic method in [24] has over 14% equal error rate when not as many training samples are used. Moreover, the authors did not investigate how robust their methods are against attackers imitating the user, which is an important issue our work investigates. The goal of [23] is similar to that of our non-critical authentication: to recognize a user out of a small number of users sharing a device. The work achieved an accuracy of about 95% only with a large number of training samples, ten versus a single one with our method, and the user must perform the given gesture in a highly constrained manner, e.g. exact timing real-time with visual feedback. These are challenging requirements for implementation on resource-constrained smart objects in mobile computing. More importantly, the gestures performed by a participant were collected from the same day, while both [24] and our work showed there were significant variations in how a user performs the same gesture over time. As a result, the result reported in [23] is likely to be overly optimistic. In contrast, our solution achieves 98% over a period of four weeks. The main reason is that our solution allows the user to create his/her personalized gesture and therefore allows more distinct features his/her gesture.

Related to our use of acceleration, the work in [25] employed accelerometers to recognize the user with the gait pattern as a behavioral biometrics. Accelerometers have also been used to solve another security-related problem, pairing of two devices [26–30]. The approach is to produce a time series of acceleration as the shared secret between two devices. Such work, however, is very different from ours in their goal and scope.

3. uWave algorithm design

In this section, we present the key technical components of uWave: acceleration quantization, dynamic time warping (DTW), and template adaptation. The premise of uWave is that *human gestures can be characterized by the time series of forces*

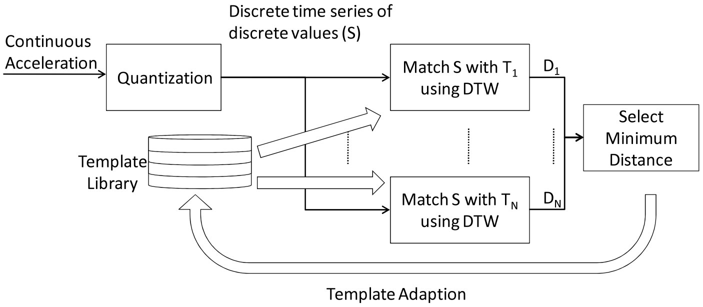


Fig. 1. uWave is based on acceleration quantization, template matching with DTW, and template adaptation.

Table 1
uWave quantizes acceleration data in a non-linear fashion before template matching.

Acceleration data (a)	Converted value
$a > 2g$	16
$g < a < 2g$	11–15 (five levels linearly)
$0 < a < g$	1–10 (ten levels linearly)
$a = 0$	0
$-g < a < 0$	-1 to -10 (ten levels linearly)
$-2g < a < -g$	-11 to -15 (five levels linearly)
$a < -2g$	-16

applied to the handheld device. Therefore, uWave bases the recognition on the matching of two time series of forces, measured by a single three-axis accelerometer.

For recognition, uWave leverages a *template library* that stores one or more time series of known identities for every vocabulary gesture, often input by the user. Fig. 1 illustrates the recognition process. The input to uWave is a time series of acceleration provided by a three-axis accelerometer. Each time sample is a vector of three elements, corresponding to the acceleration along the three axes. uWave first quantizes acceleration data into a time series of discrete values. The same quantization applies to the templates too. It then employs DTW to match the input time series against the templates of the gesture vocabulary. It recognizes the gesture as the template that provides the best matching. The recognition results, confirmed by the user as correct or incorrect, can be used to adapt the existing templates to accommodate gesture variations over time.

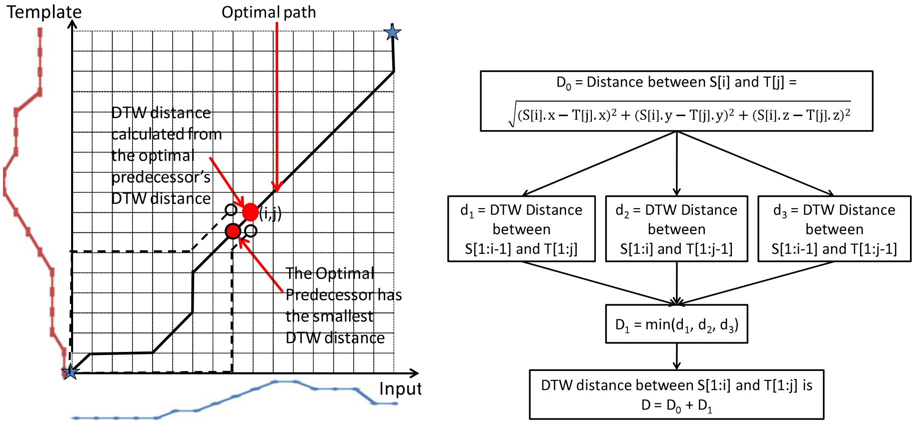
3.1. Quantization of acceleration data

uWave quantizes the acceleration data before template matching. Quantization reduces the length of input time series for DTW in order to improve computation efficiency. It also converts the accelerometer reading into a discrete value thus reduces floating point computation. Both are desirable for implementation in resource-constrained embedded systems. Quantization improves recognition accuracy by removing variations not intrinsic to the gesture, e.g. accelerometer noise and minor hand tilt.

uWave quantization consists of two steps. In the first step, the time series of acceleration is temporally compressed by an averaging window of 50 ms that moves at a 30 ms step. That is, the first data point is the average of the acceleration generated during the first 50 ms, the second is the average of the acceleration generated during 30 ms to 80 ms, and so forth. This significantly reduces the length of the time series for DTW. The rationale behind it is that intrinsic acceleration produced by hand movement does not change erratically; and rapid changes in acceleration are often caused by noise and minor hand shake/tilt. In the second step, the acceleration data is converted into one of 33 levels, as summarized by Table 1. Non-linear quantization is employed because we find that most samples are between $-g$ and $+g$ and very few go beyond $+2g$ or below $-2g$.

3.2. Dynamic time warping

Dynamic time warping (DTW) is a classical algorithm based on dynamic programming to match two time series with temporal dynamics [6], given the function for calculating the distance between two time samples. uWave employs the Euclidean distance for matching quantized time series of acceleration, i.e. the distance between two 3-axis acceleration time samples is $\sqrt{d_x^2 + d_y^2 + d_z^2}$, where d_x, d_y and d_z are the differences in acceleration along three axes, respectively. Let $S[1 \dots M]$ and $T[1 \dots N]$ denote the two time series. As shown in Fig. 2(a), any matching between S and T with time warping can be represented as a monotonic path from the starting point $(1, 1)$ to the end point (M, N) on the M by N grid. A point along the path, say (i, j), indicates that $S[i]$ is matched with $T[j]$. The matching cost at this point is calculated as the distance between


Template
Optimal path
DTW distance calculated from the optimal predecessor's DTW distance
(i,j)
The Optimal Predecessor has the smallest DTW distance
Input
D0 = Distance between S[i] and T[j] =
$\sqrt{(S[i].x - T[j].x)^2 + (S[i].y - T[j].y)^2 + (S[i].z - T[j].z)^2}$
d1 = DTW Distance between S[1:i-1] and T[1:j]
d2 = DTW Distance between S[1:i] and T[1:j-1]
d3 = DTW Distance between S[1:i-1] and T[1:j-1]
D1 = min(d1, d2, d3)
DTW distance between S[1:i] and T[1:j] is D = D0 + D1

(a) Graphic illustration of the recursive algorithm.

(b) Algorithm for computing the DTW distance between $S[1:i]$ and $T[1:j]$.

Fig. 2. Dynamic Time Warping (DTW) algorithm.

$S[i]$ and $T[j]$. The path must be monotonic because the matching can only move forward. The similarity between S and T is evaluated by the minimum accumulative distance of all possible paths, or *matching cost*.

DTW employs dynamic programming to calculate the matching cost and find the corresponding optimal path. As illustrated in Fig. 2(a), the optimal path from (1, 1) to point (i, j) can be obtained from the optimal paths from (1, 1) to the three predecessor candidates, i.e. $(i-1, j)$, $(i, j-1)$, $(i-1, j-1)$. The matching cost from (1, 1) to (i, j) is therefore the distance at (i, j) plus the smallest matching cost of the predecessor candidates. The algorithm is illustrated in Fig. 2(b). The time complexity and space complexity of DTW are both $O(M \cdot N)$.

3.3. Template adaptation

As we will show in the evaluation section, there are considerable variations between gesture samples by the same user collected from different days. Ideally, uWave should adapt its templates to accommodate such time variations. Template adaption of DTW for speech recognition has been extensively studied, e.g. [31,32], and proved to be effective. In this work, however, we only devise two simple schemes to adapt the templates. *Our objective is not to explore the most effective adaptation methods but to demonstrate the template adaptation can be easily implemented and effective in improving recognition accuracy over multiple days.*

Our template adaptation works as follows. uWave keeps two templates generated in two different days for each vocabulary gesture. It matches a gesture input with both templates of each vocabulary gesture and take the smaller matching cost of the two as the matching cost between the input and vocabulary gesture.

Each template has a timestamp of when it is created. On the first day, there is only one training sample, or template, for each gesture. As the user input more gesture samples, uWave updates the templates based on how old the current templates are and how well they match with new inputs. We develop two simple updating schemes. In the first scheme, if both templates for a vocabulary gesture in the library are at least one day old and the input gesture is correctly recognized, the older one will be replaced by the newly correctly recognized input gesture. We refer to this scheme as *Positive Update*. The second scheme differs from the first one only in that we replace the older template with the input gesture when it is incorrectly recognized. We call this scheme *Negative Update*. Positive Update only requires the user to notify uWave when the recognition result is incorrect. Negative Update requires the user to point out the correct gesture when a recognition error happens, e.g. by pressing a button corresponding to the identity of the input sample.

4. Prototype implementation

We have implemented multiple prototypes of uWave on various platforms, including the Wii remote as shown in Fig. 3, Windows Mobile Smartphones, Apple iPhone, and the Rice Orbit sensor [33]. Our accuracy evaluation is based on the Wii remote prototype, due to its popularity and ease of use.

The Wii remote has a built-in three-axis accelerometer from Analog Devices, ADXL330 [34]. The accelerometer has a range of −3g to 3g and noise below 3.5 mg when operating at 100 Hz [35]. The Wii remote can send the acceleration data and button actions through Bluetooth to a PC in real time. We implement uWave and its variations on a Windows PC using Visual C#. The implementation is about 300 lines of code. The prototype detects the start of a gesture when the 'A' button on the Wii remote is pressed; and detects the end when the button is released. While our prototype is based on the Wii remote hardware, uWave can be implemented with any device with a three-axis accelerometer of proper sensitivity and range as are those found in most consumer electronics and mobile devices.

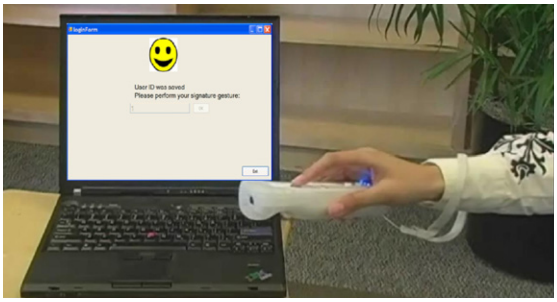


Fig. 3. Wii remote based prototype of uWave: the Wii remote sends the acceleration data through Bluetooth to the laptop that runs the recognition algorithm.

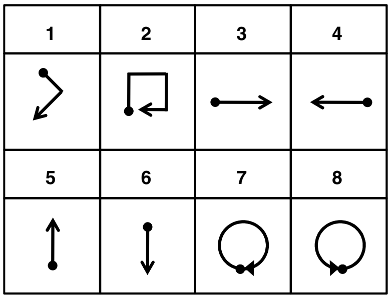


Fig. 4. Gesture vocabulary adopted from [6]. The dot denotes the start and the arrow the end.

uWave gives out recognition result without perceptible delay in our experiments based on PCs. We measured the speed of uWave implemented in C on multiple platforms. On a Lenovo T60 with 1.6 GHz Core 2 Duo, it takes less than 2 ms for a template library of eight gestures. On a T-Mobile MDA Pocket PC with Windows Mobile 5.0 and 195 MHz TI OMAP processor, it takes about 4 ms for the same vocabulary. Such latencies are too short to be perceptible to human users. We also tested uWave on an extremely simple 16-bit microcontroller in the Rice Orbit sensor [33], TI MSP430LF1611. The delay is about 300 ms. While this may be perceptible to the user, it is still much shorter than the time a gesture usually takes so that should not impair user experience.

5. Evaluation

We next present our evaluation of uWave for a vocabulary of predefined gestures based on the Wii remote prototype.

5.1. *Gesture vocabulary and database collection*

We employ a set of eight simple gestures identified by a VTT research study [4] as preferred by users for interaction with home appliances. The work also provided a comprehensive evaluation of HMM-based methods so that a comparison with uWave is possible. Fig. 4 shows these gestures as the paths of hand movement.

We collect gestures corresponding to the VTT vocabulary from eight participants with the Wii remote-based prototype. Two of them are undergraduates and others are graduate students; all but one is male. They are in 20s or early 30s, right handed.

The gesture database is collected via the following procedure. For a participant, gestures are collected from seven days within a period of about three weeks. On each day, the participant holds the Wii remote in hand and repeats each of the eight gestures in the VTT vocabulary ten times. The participants are free to hold the Wii remote in any way they want; we only ask them to keep it as consistent as possible. The database consists of 4480 gestures in total and 560 for each participant. While our participants are not demographically representative, this database provides us a statistically significant benchmark for evaluating the recognition accuracy. We have made the database open source and it can be downloaded from [36].

It is important to note that the dataset used in [4] consists of 30 samples for each gesture collected from a single user. All of the 30 samples for the same gesture were collected on the same day (the entire dataset of eight gestures were collected over two days). As we will highlight in this work, users exhibit high variations in the same gesture over the time. Samples for the same gesture from the same day cannot capture this and may lead to overly optimistic recognition results.

5.2. *Recognition without adaptation*

We first report recognition results for uWave without template adaptation.

	↙	↲	→	←	↑	↓	↻	↺
↙	92.1	0.1	2.4	1.9	0.1	2.9	0.6	0.1
↲	1.6	91.6	1.3	1.1	0.7	0.4	2.7	0.6
→	0.5	0	95.9	1.2	0.7	1.7	0	0
←	0.3	0	1.6	96.2	0.7	1.1	0	0.1
↑	0.3	0	1.5	0.6	97.0	0.5	0	0.1
↓	2.4	0	2.4	2.3	1.0	91.7	0.1	0
↻	3.4	1.9	2.6	1.7	0.4	0.7	89.2	0
↺	1.1	0.6	1.7	0.9	0.8	0.7	0	94.2

	↙	↲	→	←	↑	↓	↻	↺
↙	98.4	0	0.3	0.4	0	0.4	0.3	0.2
↲	0.5	98.3	0.2	0	0.3	0.1	0.4	0.1
→	0.2	0	98.3	0.6	0.1	0.6	0.2	0
←	0.2	0	0.3	98.8	0.3	0.2	0.2	0
↑	0.4	0	0.2	0.4	98.7	0.1	0.2	0
↓	0.7	0	0.6	0.5	0.3	97.7	0.2	0
↻	0.5	0.4	0.4	0.1	0.1	0.3	98.1	0.2
↺	0.2	0.1	0.1	0.2	0	0	0.2	99.2

Fig. 5. Confusion matrices for the VTT vocabulary without adaptation. Columns are recognized gestures and rows are the actual identities of input gestures. (Left) Tested with samples from all days (average accuracy is 93.5%); (Right) Tested with samples from the same day as the template (average accuracy is 98.4%).

5.2.1. Test procedure

Because our focus is personalized gesture recognition, we evaluate uWave using the gestures from each subject separately. That is, the samples from a participant are used to provide templates and test samples for the same subject. We employ Bootstrapping [37] to further improve the statistical significance of our evaluation. The following procedure applies to each participant separately. For clarity, let us label the samples for each gesture by the order they were collected. For the *i*th test, we use the *i*th sample for each gesture from the participant to build eight templates and use the rest samples from the same participant to test uWave. As *i* is from 1 to 70 (10 times by 7 days), we have 70 tests for each participant. Each test produces a confusion matrix that shows the percentage of times how a sample is recognized. We average the confusion matrixes for the 70 tests to produce the confusion matrix for each participant.

We average confusion matrixes of all eight participants to produce the final confusion matrixes. Fig. 5 (Left) summarizes the recognition results of uWave over the database for the VTT gesture vocabulary. In the matrixes, columns are recognized gestures and rows are the actual identities of input gestures.

uWave achieves an average accuracy of 93.5%. Fig. 5 (Left) also shows that gesture 1, 2, 6 and 7 have lower recognition accuracy in that they involve similar hand movement as each other, e.g. both gesture 1 and gesture 6 are featured by waving down movement. A closer look into the confusion matrixes for each participant reveals a large variation (9%) in recognition accuracy among different participants. *We observed that the participant with the highest accuracy performed the gestures in larger amplitude and slower speed compared to other participants.*

Our evaluation also shows the effectiveness of quantization, i.e., temporal compression and non-linear conversion, of the raw acceleration data. Temporal compression speeds up the recognition by more than nine times without a negative impact on accuracy; and non-linear conversion improves the average accuracy by 1% and further speeds up the recognition.

5.2.2. Evaluation using samples from the same day

To highlight how gesture variations from the same user over multiple days impact the gesture recognition, we modify the test procedure above so that when a sample is chosen as the template, uWave is tested only with other samples collected in the same day. Fig. 5 (Right) summarizes the recognition results averaged cross all eight participants. It shows a significantly higher accuracy (98.4%) than that of using samples from all different days. We further test the templates with testing samples collected one to six days away. The accuracy drops from 97% to 88% when the time distance increases from one day to six days.

The difference in accuracy highlights the possible variations for the same gesture from the same user over multiple days and the challenge it poses to recognition. This also indicates that the results reported by some previous work, e.g. [4,13], could be overly optimistic because the evaluation dataset was collected over a very short time.

The same-day accuracy of 98.4% by uWave with one training sample per gesture is comparable to HMM-based methods with 12 training samples (98.6%) reported in [4]. It is worth noting that the accelerometer in Wii remote provides comparable accuracy but larger acceleration range ($-3g$ to $3g$) than that used in [4] ($-2g$ to $2g$). In reality, however, the acceleration produced by hand movement rarely exceeds the range from $-2g$ to $2g$. Hence, the impact of difference in the accelerometers on the accuracy should be insignificant.

5.2.3. Tradeoff between recognition accuracy and rejection rate

A simple rejection procedure can help uWave achieve higher accuracy with a reasonably low rejection rate. The basic uWave algorithm recognizes the input gesture as the template with the smallest matching cost no matter how much the

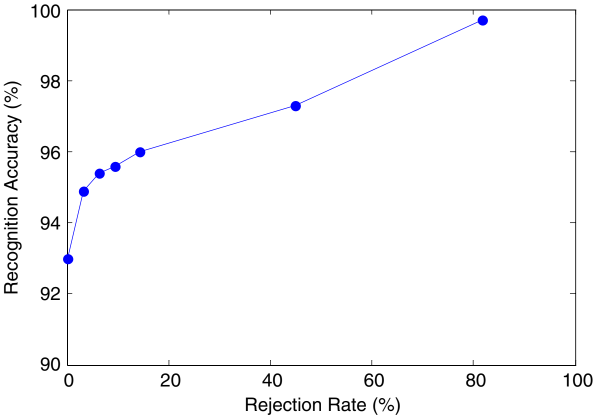


Fig. 6. Tradeoff between recognition accuracy and rejection rate.

absolute matching cost is. Such a design may produce inaccurate results when the input gesture is not similar to any template. A simple solution is to reject the input gesture, i.e. the recognition result is "unknown", if the matching costs between the input gesture and all the templates are larger than a threshold. The smaller the threshold is, the more input gestures may get rejected and the higher recognition accuracy for the input gestures not rejected. It is important to note that "rejection" is widely used in pattern recognition, e.g. speech recognition based on DTW [38]. What we adopt is the simplest method based on a pre-calculated threshold although there are more sophisticated rejection procedures that may yield better tradeoff.

Fig. 6 illustrates the tradeoff between cross-day recognition accuracy and rejection rate when the threshold varies from 5 times to 50 times of the matching cost between the input gesture and a still state acceleration sequence with the same duration. When the rejection is around 5%, the recognition accuracy is improved to 95%, compared with 93.5% without rejection.

5.3. Recognition with adaptation

The considerable difference between Fig. 5 (Left) and Fig. 5 (Right) motivates the use of template adaptation to accommodate variations over the time in order to achieve accuracy close to that in Fig. 5 (Right). We report the results next.

Again, we evaluate uWave with adaptation for each participant separately. Because the adaption is time-sensitive, we have to apply Bootstrapping in a more limited fashion. Let us label the days in which a participants' gestures were collected by the time order, from one to seven. For the ith test, we assume the evaluation starts on the ith day and applies the template adaptation in the following days, from $(i+1)$th to 7th and then from 1st to $(i-1)$th. We have seven tests for each participants and each produces a confusion matrix. We average them to produce the confusion matrix for each participant and average the confusion matrixes of all participants for the final one.

It shows an accuracy of 97.4% for Positive Update and 98.6% for Negative Update, significantly higher than that without adaptation (Fig. 5 Left) and close to that tested with samples from the same day (Fig. 5 Right). While template adaptation requires user feedback when a recognition error happens, the high accuracy indicates that it is needed only for 2%–3% of all the test samples.

6. Gesture-based 3D mobile user interface

In the next two sections, we present two applications of uWave. One of the strengths of uWave is that it can recognize three-dimensional hand movement. It has been shown that it is intuitive and convenient to navigate a 3D user interface with 3D hand gestures [39]. Qualitatively, being able to manipulate a 3D interface using a 3D gesture is much more compelling than traditional button-based solutions. In order to explore this, we developed a 3D-mobile application and integrated uWave with it to enable gesture-based navigation.

The 3D application was built around a social networking-based video-sharing service under development within Motorola. The interface shows a rotating ring that contains thumbnails of various users (a friends list) as in Fig. 7. Additionally, upon selecting a particular user, one can scroll through different video clips that have been submitted by that user. We employed uWave to navigate this user interface using a series of specific movements such as tilting and slight shaking, which are more appropriate for a mobile device when the user is focused on the screen. We also added the personalization features of uWave to allow users to re-map gestures to their liking, enabling custom navigation of the 3D interface.

The application runs on a Smartphone with its own embedded accelerometer, and is implemented in C++ for the Windows Mobile 6 Platform. The 3D interface is built and rendered using the Mobile 3D Graphics (M3G) API. The acceleration data is

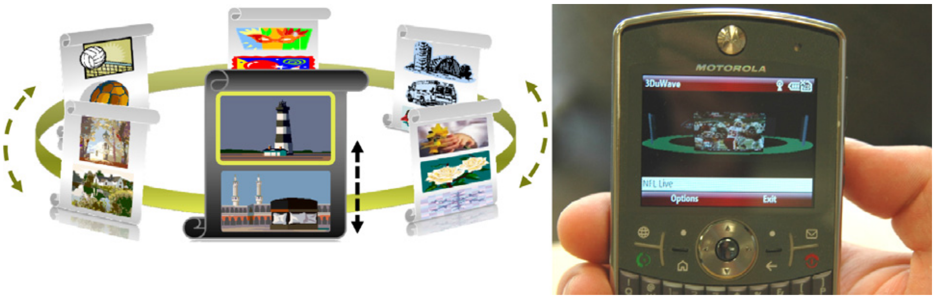


Fig. 7. Mobile 3D User Interface with uWave-based gesture interaction: (left) Illustration of the user interface and (right) prototype implementation.

sampled at 100 Hz. Even when the 3D rendering consumes a significant amount of memory, uWave works smoothly with it, without introducing any human perceptible performance degradation.

7. Gesture-based user authentication

uWave provides an interesting opportunity for gesture-based user authentication, which is lightweight in terms of computing, form factor, and user engagement. For example, a user can "shake" a phone in a particular way to log in or a TV remote to load personalized data. While many paradigms exist for user authentication, including password [19], biometrics [16–18], speech [20], and handwriting [21], accelerometer-based gesture recognition has its unique value for user authentication because of its low cost, high efficiency, and no change to the form factors of the device. These properties make it highly suitable for implementation on resource-constrained devices, e.g. mobile phones and TV remotes.

We conduct a series of user studies to investigate the feasibility and usability of using uWave for such gesture-based authentication. During our evaluation, we distinguish two different objectives of user authentication. For privacy-insensitive data, the objective of user authentication is to retrieve user-specific data instead of protecting them, e.g. personal profiles or personalized configurations on a TV remote shared by family members. In this case, accuracy and usability are dominating concerns. We call such user authentication *non-critical* and call the gestures *ID gestures*. On the other hand, there is also privacy-sensitive data, e.g. personal contacts stored in a mobile phone. In this case, the objective of user authentication is to protect the data against possible unauthorized access, or *attacks*. Therefore, resilience to attack and usability are dominating concerns. In contrast to non-critical authentication, we call such authentication *critical* and the gestures *password gestures*.

uWave plays different roles in non-critical authentication and critical authentication. In non-critical authentication, uWave identifies the best matching template from multiple templates created by different users and recognize the user as the identity behind the best-matching template. In critical authentication, uWave functions as a classic binary classifier. It calculates the matching distance between the input gesture and the template gesture representing the claimed user identity. If the matching distance is lower than a certain threshold, the input gesture is accepted as the claimed identity and otherwise rejected.

For critical authentication, uWave may make two types of error: false positive when it accepts an attacker's input; and false negative when it rejects the owner's input. A higher threshold leads to more false negatives (rejecting authentic user's gestures) so that less usable but less false positives (accepting attackers' gestures) so that it is more secure. By varying the threshold, one can obtain the receiver operating characteristic (ROC) curve, which quantitatively represents the classifier's tradeoff between false positives and false negatives. An ideal classifier should be able to achieve both low false positive and low false positive rates. In our implementation, the threshold for critical authentication is a portion of the *base distance*, which is the matching distance between the pre-recorded password gesture template and a still state acceleration sequence.

7.1. Non-critical user authentication

We aim to answer the following research questions for non-critical authentication.

- What accuracy can uWave system achieve in recognizing users based on user-created ID gestures?
- How usable is it? In particular, how challenging is it to memorize and perform an ID gesture, in comparison to conventional text ID-based authentication?
- Since users are allowed to create their own gestures, what constraints in ID gesture selection can be employed to improve the accuracy and usability?

7.1.1. Procedure

Participants and training

We recruit 25 participants. They are undergraduate and graduate students from multiple universities in the USA, aged 18 to 32, 18 males and 7 females. They major in various disciplines, including Chemistry, History, Electrical Engineering, Computer Science, Mechanical Engineering, and MBA. Some have international background.

Table 2
Procedure variations with Group A to E in the user study for non-critical authentication.

Group	Gesture selection constraint	Session frequency	Duration (week)	Textual ID
A	Collectively	Every day	1	No
B	Individually; No constraint; no rejection procedure	Every day in the first week, Every two days in the second week; Every three days in the last two weeks	4	Yes
C	Individually; have constraint; no rejection procedure	Same as Group B	4	Yes
D	Individually; have constraint and rejection procedure	Same as Group B	4	Yes
E	Individually; have constraint and rejection procedure	Every day	1	No

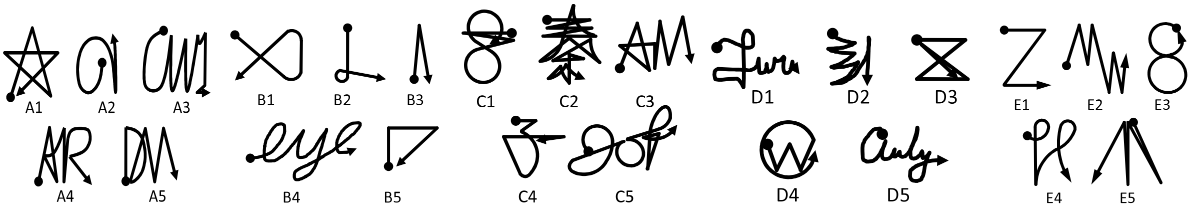


Fig. 8. ID gestures for non-critical authentication by all 25 participants.

We break the participants into five 5-person groups, called A to E in the following. We conduct the user study for each group using a similar procedure with controlled variations. Table 2 summarizes procedural difference for all five groups. Before the user study, participants are given instructions on how to use the Wii remote-based uWave prototype and are also provided with basic information regarding its template-matching mechanism. They all play with the prototype to get acquainted.

Selecting ID gestures

Table 2 summarizes the different procedures and constraints in ID gesture selection for each group.

All participants in Group A attend the first session at the same time and are asked to agree on a set of gestures as their IDs for authentication. We suggest them not to choose simple gestures as shown in Fig. 4; we further suggest the gestures to be shorter than five seconds. The participants are allowed to test the system to see whether it could recognize their selections or not: each of them input his/her gesture for a few times and the system gives them the recognition result immediately each time. This is designed to allow the participants to evaluate their collective choice of gestures. In case two gestures are highly confusing, they might have a chance to try new ones. Nevertheless, the first choices by all five participants are recognized with 100% accuracy in the session. As a result, they converge on the selection with only one attempt.

Participants in Groups B to E select their gestures one after another, without knowing the choices of others in the same group. Such a scenario is common for shared devices with gradual user adoption and provides an important alternative to the collective selection in Group A.

In order to compare the usability of ID gestures to that of commonly used textual IDs, we also ask participants in Groups B to D to choose a textual ID from a given list at the beginning of the study. Each textual ID on the list is comprised of a common used word of 3 to 8 letters and a randomly generated digit from 0 to 9. It is important to note that the purpose of these textual IDs is to provide a consistent base for usability comparison, instead of emulating the textual IDs used in real life.

For Groups B to E, we explore the impact of gesture selection constraints. The participants in Group B are free to choose any gesture as ID gesture. Four of them select very simple ones, similar to the VTT gestures. Those in Groups C to E are suggested to choose gestures more complex than the VTT gestures as shown in Fig. 4. In addition, participants in Groups D and E have to compose gestures that can pass a rejection procedure: the first participant in a group is allowed to pick up any ID gesture; the rejection procedure will reject any subsequent ID gesture choices if the matching distance between them and existing ID gestures is below 50% of the average distance between each pair of the template gestures from Group A.

One input of the selected ID gesture by each participant is saved as his/her template. We ask the participants to note down their gestures on paper and Fig. 8 shows their choices. It is important to note that the gestures are generated by six-degree free-space movement and Fig. 8 only provides the planar representation.

Collecting gestures for evaluation

The gesture collection spans over one week for Group A and E. On each day, we invite the participants back to the lab to verify themselves with their ID gestures. Each participant performs his/her ID gesture ten times and gets the recognition result immediately after each input, similar to what would happen with user recognition in reality. With the immediate feedback, the participant can adjust the next input in case of a recognition error.

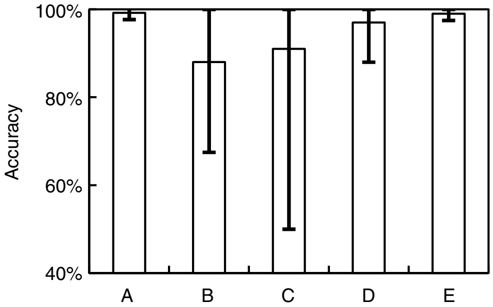


Fig. 9. Recognition accuracy for non-critical authentication: range and average for Groups A to E.

The gesture collection for Groups B, C and D takes four weeks each. We invite participants back periodically with decreasing visit frequency to study the challenge of memorizing ID gestures. Participants visit us every day in the first week, every two days in the second week, and every three days in the last two weeks. In each visit, the participants perform their ID gesture 10 times and type in their textual IDs to login a laptop. They get the authentication result (success/fail) immediately after each input. If the accumulative accuracy of a participant's inputs drops below 50% in one session, we replace his/her template using the latest input. Such template replacement is motivated by findings from Section 5.3 that it helps the uWave system cope with gesture variations over time.

Surveys

We conduct a structured survey with participants at the end of the study to evaluate their subjective opinions on the usability of gesture-based authentication, which engenders two unique tasks: (1) memorizing the ID gesture and (2) performing it.

Our hypothesis to compare the difficulties of memorizing a gesture and a textual ID is:

H1: memorizing an ID gesture is as difficult as or more difficult than memorizing the pre-composed textual ID.

Our hypothesis to compare the physical difficulty of performing a gesture and typing in a textual ID is:

H2: performing an ID gesture is as difficult as or less difficult than typing in a pre-composed textual ID.

In the survey, the participants are asked to rate (1) the difficulty of memorizing the gesture and the user ID in a 0 to 10 scale; (2) the difficulty of performing the gesture and typing in the textual ID in a 0 to 10 scale; (3) their agreement with two statements: "Memorizing a gesture is no more challenging than memorizing a textual ID" and "Performing a gesture is no more physically challenging than typing in a textual ID". There are also open-ended questions that ask them to explain why.

We are aware that the selection of textual IDs in our experiment setting is different from most real-life scenarios and we do not try to compare uWave with a practical authentication method. Our purpose is to have a consistent benchmark (textual IDs with similar structure and length) to compare the gesture-based method with.

7.1.2. Authentication results

Fig. 9 shows the average recognition accuracy in the first week for the five groups, since the data collection procedures in the first week are the same for all five groups. uWave achieves an average accuracy of 99.2% for Group A in which gesture complexity constraint is suggested and participants collectively select their ID gestures. Participants in Groups B to E select their ID gestures without knowing their peers'. The difference in the constraints of their gesture selection is detailed in Table 2. From Groups B to E, we observe average accuracy increased from 88% to 99% due to gesture complexity constraints and the rejection procedure, which will be explained below.

Selection constraints improve accuracy

The groups with complexity constraint and rejection procedure outperform the others: Group C has higher accuracy than Group B because of the complexity constraint; Groups D and E beat Group C, thanks to the rejection procedure. The complexity constraint eliminates simple gestures which can be easily confused with each other; the rejection procedure guarantees enough difference between gesture templates. Both of these two conditions are important to help participants create usable ID gestures.

Template replacement improves accuracy

As shown in Section 5.3, the same gestures performed by the same participant had significant variations over time. Thus template replacement is important to adapt to such variations. From the non-critical authentication study, we also observe that template replacement when the accuracy drops below certain threshold (50% in our study) can considerably improve accuracy. It enables users to overcome poorly input templates and adapt to performance variations over long time. In our study, B1 and B5 have low accuracy at the first three to five sessions, get their templates replaced when the accuracy drops below 50%, and achieve 92% and 98% accuracy respectively afterward. B3 has 50% accuracy at the very beginning but maintains 100% for the next a few sessions after template replacement. C2 and C5 have low accuracy (20%–60%) in the first week, get templates replaced in the second week, and achieve 100% afterwards.

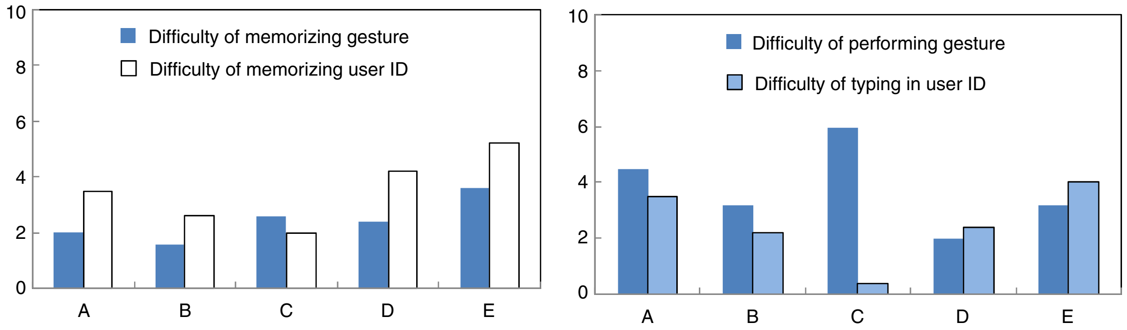


Fig. 10. Survey results for difficulty of memorizing (left) and performing (right) a gesture for Group A to E.

7.1.3. *Usability evaluation*

We next evaluate the usability of gesture-based non-critical authentication in terms of difficulties in memorizing and performing an ID gesture. Fig. 10 shows the group-wise average difficulty ratings. We analyze the ratings through hypothesis testing below. We are aware of the debate on whether data from Likert scales should be viewed as interval-level data or ordered-categorical data [40]. In our survey, however, a visual analog scale with equal spacing between responses is shown to the participants and more than five levels are provided. Therefore, we believe it is propitiate to use a parametric statistical test for analysis.

How difficult is memorizing gestures?

We use *independent two-sample* t-test to analyze the survey results. With data from all 25 participants, there is a significant difference in the means of the difficulty rating for memorizing gesture and textual ID ($P = 0.04$). Since the P-value is smaller than our significance level (5%), we reject H1 as stated in Section 7.1.1, accept its alternative hypothesis, and conclude that memorizing a gesture is less difficult than a pre-composed textual ID.

How difficult is performing gestures?

We analyze the difficulty of performing gestures in a similar way with an independent two-sample t-test. The result ($P = 0.24$) shows there is not enough evidence to reject H2 as stated in Section 7.1.1. In other words, our study does not find enough evidence to support that performing a gesture is more difficult that typing in a textual ID.

We hypothesize that participants' difficulty rating is negatively correlated to the recognition accuracy. For example, the participants in Group C consider performing a gesture much more difficult than typing in a textual ID. At the same time, Group C exhibits the largest variance in recognition accuracy as well as low average accuracy. To understand the correlation between accuracy and difficulty rating, we calculate the correlation coefficient between recognition accuracy and difficulty rating of performing a gesture for all five groups. The correlation coefficient of -0.45 shows medium correlation [41] between accuracy and difficulty rating, meaning that the higher accuracy a user achieves, the lower difficulty he/she is likely to rate performing a gesture.

For Group D and E who receive both complexity constraint and the rejection procedure for ID gesture selection, the difficulty of performing ID gestures is perceived as similar to that of typing in a pre-composed textual ID.

7.2. *Evaluation of critical user authentication*

Critical authentication is aimed to guard privacy-sensitive data from unauthorized access. It is important to note that we do not expect gesture-based authentication to provide strict security but consider it as a convenient lightweight authentication method which can be combined with traditional methods for enhanced security. For example, if the device receives several false gestures, it will activate password protection. We next explore whether uWave can recognize an owner-created gesture reliably while withstanding malicious forging, or *attack* as we referred to in the rest of the paper.

7.2.1. *Objectives*

We seek to answer the following questions through user studies.

- What tradeoffs between security and usability can the uWave-based solution achieve?
- How security can be jeopardized if the attacker sees the owner's gesture performance, which can be much more visible than textual password entry?

7.2.2. *Procedure*

Participants

We recruit ten participants, three females and seven males, aged from 20 to 32. Nine of them are graduate students and one is an undergraduate student. Their majors include Electrical Engineering, Computer Science, Psychology, Physics,

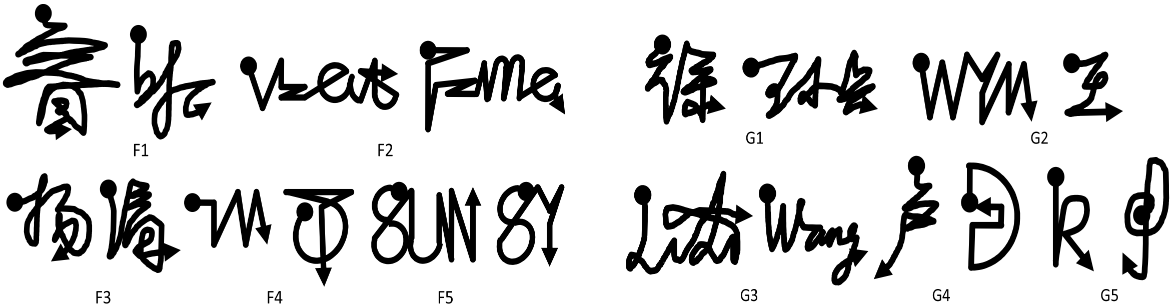


Fig. 11. Password gestures for critical authentication (two for each participant).

Applied Mathematics, and Bio-engineering. We assign them to two five-person groups, F and G, in order to study the impact of visual disclosure.

Tasks of the participants

In the study, a participant verifies himself/herself with his/her password gestures and attempts to forge his/her group peers' password gestures. A participant is called the *owner* of his/her own password gestures but called *attacker* when he/she tries to forge the password gestures from others. The only difference between Groups F and G is that attackers in Group F do not see the owner performing the password gestures; attackers in Group G do see it through a video recording, which we call *visual disclosure*. For visual disclosure, the recording camera faces the front of the performers for all password gestures.

The study takes five days. On the first day, each participant selects two password gestures, each for one form of recognition feedback as explained later. In the following four days, the participant comes back for two tasks: (1) to perform their own password gestures for five times to verify themselves; (2) to forge the password gestures of other participants in the same group for five trials; once the attacker has the first successful attack, he/she will have five extra trials after that. One participant attacks a different victim on each day. As a result, each password gesture is attacked for at least 20 times by four participants. Fig. 11 shows the gesture selections of Group F and G.

It is important to note that we choose to allow only five trials in forging a password gesture by an attacker because the system can resort to a more reliable authentication method, such as a conventional password, when several attempts have failed. Such a paradigm has already been widely used in other forms of authentication.

Two forms of authentication feedback

We also study the effect of different forms of recognition feedback by providing the authentication results in two forms: success/fail and matching distance. Each participant has two password gestures. When the first password gesture is verified by the owner or attacked by an attacker, the authentication feedback is whether he/she succeeds or fails. The recognition result is based on a predetermined default threshold, calculated as a quarter of the base distance (defined in Section 7). The trials with success/fail feedback are used to generate the baseline performance with the fixed threshold. For the second password gesture, the feedback is the matching distance between the input gesture and the gesture template of the owner recorded at the first day of the study. With the matching distance feedback, a participant knows whether he/she is getting closer or not. The trials with matching distance feedback are used for ROC analysis with various thresholds.

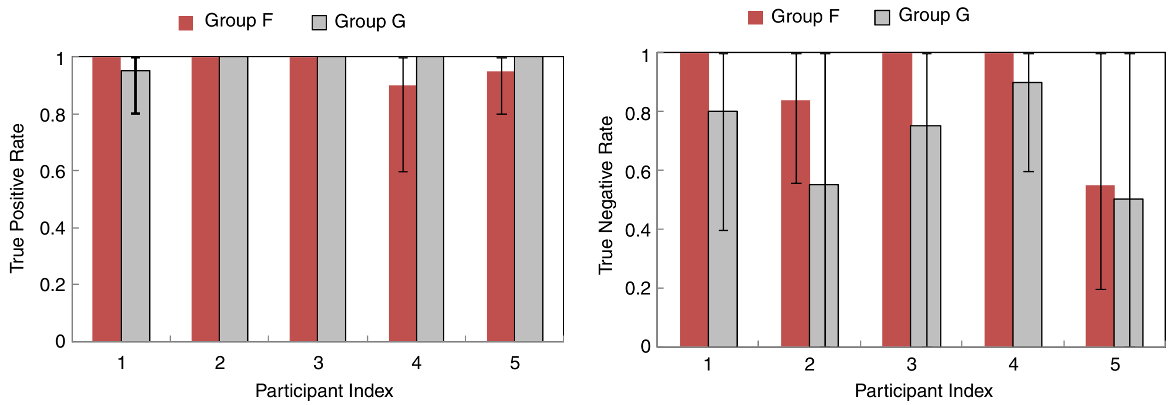
7.2.3. Authentication results

Experiments with Group F demonstrate that uWave can achieve state-of-the-art false positive rate and false negative rate when the attacker does not see the target gesture. Not surprisingly, attackers in Group G encounter a higher false positive rate due to visual disclosure. With a closer look into the results with matching distance feedback, however, it is possible to achieve both high usability and security for both Groups F and G if the rejection threshold can be set individually for each owner.

Baseline performance with default threshold

Recall that the default threshold in trials with success/fail feedback is set as a quarter of the base distance. Fig. 12 presents the results from this default threshold, where the true positive rate equals one minus false negative rate and the true negative rate equals one minus false positive rate. As in Fig. 12 (left), results for trials with success/fail feedback show that uWave can correctly recognize the input gesture all the time for all participants except F5 and G1. F5 and G1 each have two and one false negatives, respectively. Therefore, the rejection threshold based on a quarter of the base distance leads to 98% and 99% average true positive rates for Groups F and G, respectively.

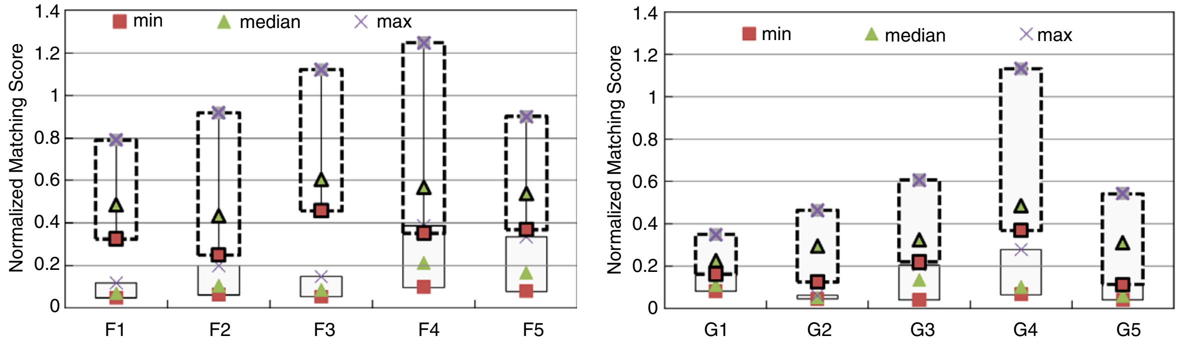
As to security as shown in Fig. 12 (right), all forged gestures are correctly rejected for all Group F participants except F2 and F5. uWave falsely accepts one to four of the faked gestures from other participants as they attack the password gestures of F2 and F5: F1 has one successful attack on F2 but fails to repeat it in the following five trials; F5 has the first success targeting F2 in the fourth trial and achieves four successful attacks among five additional trials. F1, F2, and F3 have two to four attack successes against F5. Overall, the same rejection threshold leads to 88% true negative rate.



(a) True positive rate.

(b) True negative rate.

Fig. 12. Baseline performance for each owner of Group F and G with success/fail feedback.



(a) Group F.

(b) Group G.

Fig. 13. Matching distance of the owners (regular outline) and the attackers (dashed outline). The boxes indicate the range.

With the same threshold, true negative rates are significantly lower in Group G in which attackers were given visual disclosure. The average is 70% for Group G versus 88% for Group F. It is important to note that their true positive rates are different too. Therefore, the difference between their true negative rates should not be interpreted as 88 − 70 = 18%, as we will see later in ROC analysis.

How close are attackers?

For trials with matching distance feedback, Fig. 13 shows the statistics of matching distances per participant in the form of box plots. The distances are normalized by the base distances of each password gesture. It is important to note that each participant uses two different password gestures for trials with success/fail feedback and with matching distance feedback. For Group F, the matching distances by attackers are always higher than those by the owner. It means if the threshold of rejection is carefully selected for each owner, it is possible to achieve 100% true positive rate and true negative rate for all owners except F4. If the proper threshold for an owner can be learned over time from multiple input samples, the performance can be significantly better.

For Group G in which visual disclosure is given to attackers, Fig. 13(b) shows a non-trivial difference between the matching distances by the owner and those by the attackers. Despite the low true negative rate with 0.25 as rejection threshold, matching distances by attackers are higher than those by the corresponding owner, similar to Group F. Hence it is still not trivial for attackers to forge a password gesture even with visual disclosure.

ROC analysis

Although it is possible to tune the rejection threshold for each user individually, an understanding of how uWave performs with a common threshold for all users is still important. To illustrate the tradeoffs between true positive and false positive in this case, Fig. 14 presents the receiver operating characteristics (ROC) curves for Groups F and G. We calculate the average true positive rates and false positive rates on all participants in each group by varying the rejection threshold from 0 to 0.5.

The ROC curve can help us decide a common threshold for all owners to achieve different tradeoff between false positive and true positive. A threshold between 0.15 and 0.2 will deliver nearly 95% true positive rate and below 2% false positive

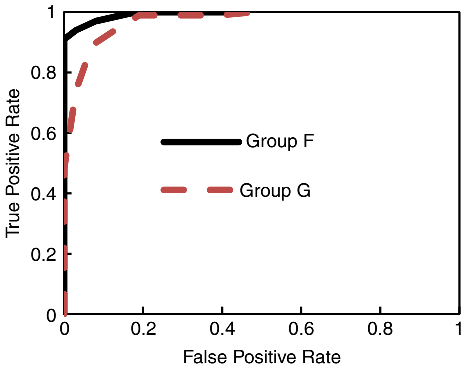


Fig. 14. ROC curves of the uWave-based critical authentication.

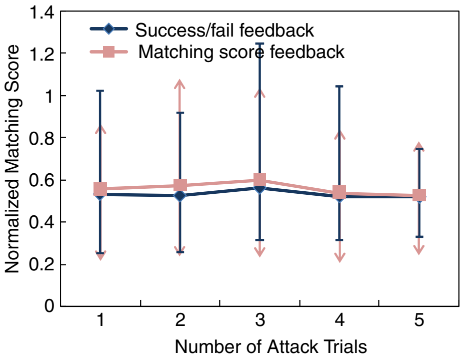


Fig. 15. Normalized matching distances over multiple consecutive attempts.

rate for Group F and 90% true positive and 5% false positive rate for Group G. Using the ROC curve, we can also estimate the equal error rate (when false positive rate and false negative rate is the same) as 3% for Group F and 10% for Group G.

Impact of visual disclosure

Not surprisingly, our study shows that visual disclosure increases false positives. As shown in Fig. 14, under the same true positive rate, the false positive rate of Group G can be up to 10% higher than that of Group F. Such high false positive rate is likely to make the proposed authentication method useless, even for applications that do not require strict security.

Impact of feedback

To explore the impact of different forms of feedback, we calculate the average matching distances of the attackers from the first trial through the fifth trial and present them in Fig. 15. We make two observations. First, there is no clear trend in the attackers' performance as the number of trials increases. Second, there is no meaningful difference between the matching distances of success/fail feedback and those of matching distance feedback. We conjecture that the time series of acceleration is very complex and the space of exploration is simply too large to explore blindly, even with the feedbacks. As a result, even if the attackers know how close they are, it is still challenging for them to improve their attack.

8. Discussion

We next address the limitations of uWave and gesture recognition based on accelerometers in general.

8.1. Gestures and time series of forces

Due to a lack of a standardized gesture vocabulary, human users may have diverse opinions on what constitutes a unique gesture. As noted early, the premise of uWave is that human gestures can be characterized as time series of forces applied to a handheld device. Therefore, the temporal dynamic of gestures is closer to speech than to handwriting, which is usually recognized as the final contours without regard to the time sequence of the contours. However, it is important to note that while one may produce the three-dimensional contour of the hand movement given a time series of forces, the same contour may be produced by very different time series of forces. Nevertheless, our evaluation gesture samples were collected without enforcing any definition of gestures to our participants. The high accuracy of uWave indicates that its premise is close to how users perceive gestures and how users perform gestures.

8.2. Challenge of tilt

On the other hand, uWave relies on a single three-axis accelerometer to infer the force applied. However, *the reading of the accelerometer does not directly reflect the external force*, because the accelerometer can be tilted around three axes. The same external force may produce different accelerations along the three axes of the accelerometer if it is tilted differently; likewise, the different forces may also produce the same accelerometer readings. Only if the tilt is known, the force can be inferred from the accelerometer readings.

The opportunity for detecting the tilt during hand movement is very limited with a single accelerometer. We have tried two methods to no avail. The first method assumed a relatively constant tilt during gesture performance due to different ways in holding the device. It transforms the whole time series of acceleration, assuming the tilt angle. It generates multiple time series assuming multiple possible tilt angles. For example, to test a sample A against a template B, we transform A to A1 and A2 assuming the device is tilted by $\pm 10^\circ$ along the y-axis, respectively. Then we then compare A and its two transformations, A1 and A2, to B and select the one that matches best. The second method does not assume a constant tilt. It deviates from the basic DTW algorithm by allowing each pair of matching points on the DTW grid (See Fig. 2) to have multiple DTW distances with different tilting angles. In other words, the DTW grid became three dimensional with the tilting angle as the third dimension. Unfortunately, while both methods help with matching samples of the same gesture collected with different tilts, they also increase the confusion between certain gestures, largely due to the confusion between gravity and the external force. When the device is tilted, the angle of the external force changes accordingly while gravity stays constant. Accelerometer-only methods, as described above, are incapable of effectively differentiating the external force and gravity. To fully address tilt variation, extra sensors, e.g. compass and gyroscope, will be necessary.

8.3. User-dependent vs. User independent recognition

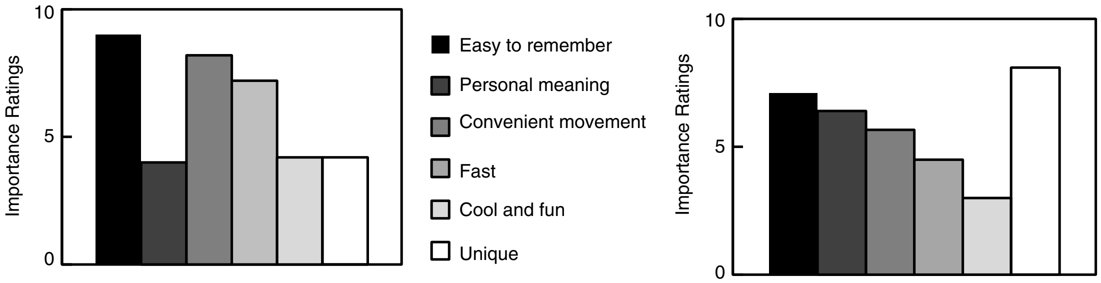
This work and numerous others are targeted at user-dependent gesture recognition only. The reasons are multiple. First, user-independent gesture recognition is difficult. Our database shows great variations among participants even for the same predefined gesture. For example, if we treat all the samples in the database as from the same participant and repeat our bootstrapping test procedure, the accuracy will decrease to 75.4% from 98.4% for user-dependent recognition. To improve the accuracy of user-independent recognition, a large set of training samples and a statistical method is necessary. More importantly, research is required to identify the common "features" from the acceleration data for the same gesture. In speech recognition, MFCC and LPCC have been found to capture the identity of speech very effectively. Unfortunately, we do not know their counterparts for acceleration-based gesture recognition. Second, user-independent gesture recognition may not be as attractive as speaker-independent speech recognition because there are no standard or commonly accepted gestures for interaction. Commonly recognized gestures by humans are often simple, such as those in the VTT vocabulary. As they are short and simple, however, they can be easily confused with each other, in particular with the presence of tilt and user variations. On the other hand, for personalized gestures composed by users, it is almost impossible to collect a large dataset for statistical methods to be effective.

8.4. Gesture vocabulary selection

The confusion matrixes presented in Fig. 5 highlight the importance of selecting the right gesture vocabulary for higher accuracy. As from Fig. 5, we can see that uWave often confuses Gesture 1 with Gesture 7. The reason is that tilt of the handheld device can transform different forces into similar accelerometer readings. Unlike speech recognition, gesture recognition has more flexible inputs, because the user can compose gestures without the constraint of a "language". More complicated gestures may lead to higher accuracy because they are likely to have more features that distinguish them from each other, in particular, offsetting the effect of tilt and gravity. Nevertheless, complicated gestures pose a burden to human users: the user has to remember how to perform complicated gestures in a consistent manner and associate them with some unrelated functionality. Eventually, the number of complicated gestures a user can comfortably command may be quite small. This may limit gesture-based interaction with a relatively small vocabulary for which uWave indeed excels.

It is interesting to note how our participants compose their ID and password gestures for user authentication. First, the selected gestures are very symbolic, such as regular shapes, letters, and characters in the native languages of the participants. Unlike speech or handwriting for which we are familiar with a well defined vocabulary, gestures are not employed in our everyday life for human to computer interaction so that lack of a defined vocabulary commonly recognized by users. Therefore, gestures based on well-known concepts and symbols are easier to memorize as well as to perform consistently. Second, the selected gestures often carry personal meanings. For example, some of them are the name initials of the participants. Such choice provides an easy solution for the uniqueness of gestures that can be easily memorized. Third, not surprisingly, password gestures for critical authentication from Groups F and G are significantly and consistently more complicated than ID gestures for non-critical authentication from Groups A to E.

For non-critical authentication, we note that uWave works well for both collective and individual procedures of ID gesture selection. That it works well for collectively selected gestures implicates that uWave distinguishes gestures in a similar way human users do; that it works well for gestures selected under uWave supervision implicates that uWave is effective in guiding users for rapidly selecting proper gestures without knowledge of others' gestures.



(a) Non-critical authentication.

(b) Critical authentication.

Fig. 16. Importance ratings of factors in gesture selection.

To further understand gesture selection, we ask the participants to rate the importance of several factors in their gesture selection in the survey. These factors include "easy to remember", "having personal meaning", "convenient for hand and arm movement", "fast to perform", "cool and fun to perform", and "likely to be unique". The average ratings are shown in Fig. 16(a). Not surprisingly, "easy to remember", "convenient for movement" and "fast to perform" are the most important three factors for non-critical authentications. All three factors are concerned with usability, memorizing and performing the gesture.

In contrast, the three most important factors for critical authentication are "unique", "easy to remember", and "personal meaning". While participants still care about "easy to remember", they consider security as more important than difficulty of performing gestures. "Unique" is rated significantly higher than in non-critical authentication, indicating that the participants consider uniqueness lead to better security. In addition, "personal meaning" also receives considerably higher rating than in non-critical authentication. When answering the open ended questions about their gesture selection, the participants in critical authentication indicate personal meanings help them remember rather complicated gestures. In contrast, participants in non-critical authentication select simpler gestures and do not need personal meanings to help them remember the gestures.

As mentioned in Section 7.1.2, we also observe that gesture selection can have a great impact on the tradeoff between security and usability for critical authentication. Our observations suggest that abrupt movement changes in gestures create fine features in the time series of acceleration and therefore can make it more challenging to forge.

8.5. Improving critical authentication

While our user evaluation shows that accelerometer-based authentication works well for non-critical authentication in terms of both usability and accuracy, it apparently cannot provide the strict security required by critical authentication in many applications. However, with an equal error rate of 3%, it is still very promising for applications in which strict security is not necessary or it can be combined with other methods to achieve an even lower rate.

As we show in Section 7.2.3, visual disclosure can potentially render the authentication method useless even for applications that do not need strict security. Therefore, visual concealment is needed. While it is difficult to prevent others from seeing one perform the gesture, one may be able to hide the starting and end points of the password gesture. This can be easily implemented on many platforms. For example, our Wii remote-based implementation requires the user to hold a button on the remote while performing a gesture. Since it is difficult for the attackers to clearly see whether the user has pressed the button or not, the user can add a spurious movement before pressing the button or after releasing the button to hide the real gesture. We also suggest employing 3D movements in the password gesture in order to make it more challenging to forge.

9. Conclusions

We present uWave for interaction based on personalized gestures and physical manipulations of a consumer electronic or mobile device. uWave employs a single accelerometer so it can be readily implemented on many commercially available consumer electronics and mobile devices. The core of uWave includes dynamic time warping (DTW) to measure similarities between two time series of accelerometer readings; quantization for reducing computation load and suppressing noise and non-intrinsic variations in gesture performance; and template adaptation for coping with gesture variation over the time. Its simplicity and efficiency allow implementation on a wide range of devices, including simple 16-bit microcontrollers.

We evaluate the application of uWave to user-dependent recognition of predefined gestures with over 4000 samples collected from eight users over multiple weeks. Our experiments demonstrate that uWave achieves 98.6% accuracy starting with only one training sample. This is comparable to the reported accuracy by HMM-based methods [4] with 12 training samples (98.9%). We show that the quantization improves recognition accuracy and reduces the computation load. Our evaluation also highlights the challenge of variations over the time to user-dependent gesture recognition and the challenge of variations across users to user-independent gesture recognition. We presented two applications of uWave: gesture-based authentication and mobile 3D interface with gesture-based navigation on an accelerometer-enhanced Smartphone. Both applications show high recognition accuracy and recognition speed with different hardware features and system resources.

We also investigate the feasibility and usability of gesture-based user authentication using uWave. For non-critical authentication, uWave recognizes the user from a small group of possible users; for critical authentication, uWave verifies the claimed user identity. We report an extensive evaluation of gesture-based user authentication with a comprehensive set of user studies. Our user studies have demonstrated that accelerometer-based gesture recognition can provide a feasible and usable solution for non-critical user authentication. For critical authentication, our uWave system achieves a state-of-the-art performance. With 3% equal error rate, it can be useful when strict security is not expected. However, we show that visual disclosure can potentially increase the equal error rate to 10%, making the authentication method useless even for non-strict security. There is a need for future research to cope with visual disclosure.

We believe uWave is a major step toward building technology that facilitates personalized gesture recognition. Its accurate recognition with one training sample is critical to the adoption of personalized gesture recognition in a range of devices and platforms and to the realization of novel gesture-based navigation of next generation user interfaces.

Acknowledgments

The authors would like to thank Crysta Metcalf and Elaine Huang for their help with our user study design. The work is supported in part by NSF awards CNS/CSR-EHS 0720825 and IIS/HCC 0713249 and by a gift from Motorola Labs. The authors would like to thank the participants in our user studies and thank the anonymous reviewers for useful feedbacks that improved the paper.

References

[1] T. Baudel, B.-L. Michel, Charade: Remote control of objects using free-hand gestures, Communications of the ACM 36 (1993) 28–35.
[2] X. Cao, R. Balakrishnan, VisionWand: Interaction techniques for large displays using a passive wand tracked in 3D, in: Proc. ACM Symp. User Interface Software and Technology, UIST, ACM, Vancouver, Canada, 2003.
[3] J.K. Perng, B. Fisher, S. Hollar, K.S.J. Pister, Acceleration sensing glove (ASG), in: Digest of Papers for Int. Symp. Wearable Computers, 1999, pp. 178–180.
[4] J. Kela, P. Korpipää, J. Mäntyjärvi, S. Kallio, G. Savino, L. Jozzo, D. Marca, Accelerometer-based gesture control for a design environment, Personal Ubiquitous Computing 10 (2006) 285–299.
[5] Y. Wu, T.S. Huang, Vision-based gesture recognition: A review, in: Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction, Springer-Verlag, 1999.
[6] C.S. Myers, L.R. Rabiner, A comparative study of several dynamic time-warping algorithms for connected word recognition, The Bell System Technical Journal 60 (1981) 1389–1409.
[7] Nintendo, Nintendo Wii, http://www.nintendo.com/wii/.
[8] G. Heumer, H.B. Amor, M. Weber, B. Jung, Grasp recognition with uncalibrated data gloves—A comparison of classification methods, in: IEEE Virtual Reality Conference, 2007, p. 19.
[9] S. Ronkainen, J. Häkkilä, S. Kaleva, A. Colley, J. Linjama, Tap input as an embedded interaction method for mobile devices, in: Proc. Int. Conf. Tangible and Embedded Interaction, ACM, Baton Rouge, LA, 2007.
[10] I.J. Jang, W.B. Park, Signal processing of the accelerometer for gesture awareness on handheld devices, in: W.B. Park (Ed.), Proc. IEEE Int. Wkshp. Robot and Human Interactive Communication, 2003, pp. 139–144.
[11] P. Keir, J. Payne, J. Elgoyhen, M. Horner, M. Naef, P. Anderson, Gesture-recognition with non-referenced tracking, in: IEEE Symp. 3D User Interfaces, 3DUI, 2006, p. 151.
[12] AiLive Inc, AiLive LiveMove Pro, http://www.ailive.net/liveMovePro.html.
[13] J. Mäntyjärvi, J. Kela, P. Korpipää, S. Kallio, Enabling fast and effortless customisation in accelerometer based gesture interaction, in: Proc. Int. Conf. Mobile and Ubiquitous Multimedia, ACM, College Park, MA, 2004.
[14] D. Wilson, A. Wilson, Gesture Recognition using XWand, Robotics Institute, Carnegie Mellon University, 2004.
[15] J.O. Wobbrock, A.D. Wilson, Y. Li, Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes, in: Proc. ACM Symp. User Interface Software and Technology, UIST, 2007.
[16] D. Maltoni, Handbook of Fingerprint Recognition, Springer, 2003.
[17] W. Zhao, R. Chellappa, P.J. Phillips, A. Rosenfeld, Face recognition: A literature survey, ACM Computing Surveys 35 (2003) 399–458.
[18] R.P. Wildes, Iris recognition: An emerging biometric technology, Proceedings of the IEEE 85 (1997) 1348–1363.
[19] B.D. Payne, W.K. Edwards, A brief introduction to ssable security, IEEE Internet Computing 12 (2008) 13–21.
[20] J.P. Campbell Jr., Speaker recognition: A tutorial, Proceedings of the IEEE 85 (1997) 1437–1462.
[21] D. Impedovo, G. Pirlo, Automatic signature verification: The state of the art, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 38 (2008) 609–635.
[22] F. Okumura, A. Kubota, Y. Hatori, K. Matsuo, M. Hashimoto, A. Koike, A study on biometric authentication based on arm sweep action with acceleration sensor, in: Int. Symp. Intelligent Signal Processing and Communications, 2006, pp. 219–222.
[23] E. Farella, S. O'Modhrain, L. Benini, B. Riccó, Gesture signature for ambient intelligence applications: A feasibility study, in: Pervasive Computing, 2006, pp. 288–304.
[24] K. Matsuo, F. Okumura, M. Hashimoto, S. Sakazawa, Y. Hatori, Arm swing identification method with template update for long term stability, in: Advances in Biometrics, 2007, pp. 211–221.
[25] J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S.M. Makela, H.A. Ailisto, Identifying users of portable devices from gait pattern with accelerometers, in: Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 2, 2005, pp. ii/973-ii/976 Vol. 2.
[26] K. Hinckley, Synchronous gestures for multiple persons and computers, in: Proc. ACM Symp. User Interface Software and Technology, UIST, ACM, Vancouver, Canada, 2003.
[27] R.M.a.H. Gellersen, Shake well before use: Authentication based on accelerometer data, in: Proc. Int. Conf. Pervasive Computing (Pervasive), 2007.
[28] S.N. Patel, J.S. Pierce, G.D. Abowd, A gesture-based authentication scheme for untrusted public terminals, in: Proc. ACM Symp. on User Interface Software and Technology, UIST, ACM, Santa Fe, NM, 2004.
[29] D. Kirovski, M. Sinclair, D. Wilson, The Martini Synch: Joint fuzzy hashing via error correction, Security and Privacy in Ad-hoc and Sensor Networks (2007) 16–30.

[30] L.E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, H.-W. Gellersen, Smart-its friends: A technique for users to easily establish connections between smart artefacts, in: Proc. Int. Conf. Ubiquitous Computing, Springer-Verlag, Atlanta, Georgia, 2001.
[31] F.R. McInnes, M.A. Jack, J. Laver, Template adaptation in an isolated word-recognition system, IEE Proceedings 136 (1989).
[32] R. Zelinski, F. Class, A learning procedure for speaker-dependent word recognition systems based on sequential processing of input tokens, in: Proc. IEEE ICASSP, 1983.
[33] Rice Efficient Computing Group, Rice orbit sensor platform, in http://www.recg.org/orbit.htm.
[34] H. Wisniowski, Analog Devices and Nintendo collaboration drives video game innovation with iMEMS motion signal processing technology, Analog Devices (2006).
[35] Analog Device, Small, low power, 3-Axis ±3g i MEMS®accelerometer: ADXL330 datasheet, 2006.
[36] Rice Efficient Computing Group, uWave Data Set, http://www.ruf.rice.edu/~jl5/research_files/uwave.htm.
[37] M.R. Chernick, Bootstrap: A Practitioner's Guide, 1999.
[38] L. Lamel, L. Rabiner, A. Rosenberg, J. Wilpon, An improved endpoint detector for isolated word recognition, IEEE Transactions on Acoustics, Speech and Signal Processing 29 (1981) 777.
[39] K. Hinckley, J. Tullio, R. Pausch, D. Proffitt, N. Kassell, Usability analysis of 3D rotation techniques, in: Proc. ACM Symp. User Interface Software and Technology, UIST, 1997.
[40] T.D. DL Clason, Analyzing data measured by individual likert-type items, Journal of Agricultural Education 35 (4) (1994) 31–35.
[41] J. Cohen, P. Cohen, S.G. West, L.S Aiken, Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences, 3rd ed, Lawrence Erlbaum Associates, Hillsdale, NJ, 2003.