Chapter 12
Improving Cell Phone Based Gait Identification with Optimal Response Time Using Cloudlet Infrastructure

Thang Hoang, Viet Vo, Chuyen Luong, Son Do and Deokjai Choi

Abstract In this paper, we propose an improved gait identification based on signal collected from mobile sensors (e.g. accelerometer, magnetometer). Based on the observation from previous works, we found that there are restrictions which could negatively affect the efficiency of the system when it is applied in reality. For example the installation error has never been considered well. Additionally, performing identification tasks on mobile devices with limited resource constraints is also a big challenge. In this paper, we propose our own identification method which achieves better accuracy than previous works by taking a deep look at processing steps in gait identification issue. Moreover, the interaction between our identification model and human interaction is improved by minimizing the time delay to perform identification. To do this, the VM-based cloudlet infrastructure is also constructed to perform assigning computation tasks from mobile to nearby powerful PCs that belong to the cloudlet. From initial experiment, the archived accuracy of our identification model was approximately 98.99 % and the response time was reduced by 95.8 %.

Keywords Behavioral biometric · Accelerometer · Pattern recognition · Gait identification · Authentication · Cloudlet infrastructure · Mobile security

T. Hoang (✉) · V. Vo · C. Luong · S. Do · D. Choi
ECE, Chonnam National University, Gwangju, South Korea
e-mail: hmthang2812@gmail.com

D. Choi
e-mail: dchoi@jnu.ac.kr

H.-K. Jung et al. (eds.), *Future Information Communication Technology and Applications*, Lecture Notes in Electrical Engineering 235, DOI: 10.1007/978-94-007-6516-0_12,

12.1 Introduction

The explosion of mobility nowadays is setting a new standard for information technology industry. Mobile devices have skyrocketed in recent years. Such device functionalities are not limited in calling, or texting sms, but providing a variety of utilities including entertainment applications (e.g. game, music, video, etc.) and businesses applications like e-commerce or m-banking.

However, considering mobile devices as safe and secure devices and placing whole trust in them could make owners face up to security issues. They can be easily stolen, or illegally accessed [1] and hence making sensitive data stored in them become more vulnerable. Consequently, identification settings have evolved to become a more priority issue. The most widely-used identification methods in mobile devices are currently PIN and password because of their ease in use and implementation. However, these methods are not always effective considering security aspects [1] and cause inconveniences since they require users to enter passkey implicitly.

Thus, a friendlier identification mechanism is desired to be found and aimed to ameliorate the mobile security. Recently, approaches based on behavioral biometric such as gait characteristic had been already implemented on wearable sensors [2–11]. Achieved results showed potential opportunities for publishing a new well-secured identification mechanism on mobile. However, there still exist restrictions related to laboratory environment that can cause impossibilities when it is applied in reality. Moreover, computing steps in gait identification on local mobile with resource constrains is such a big challenge. The time delay is increased significantly because mobile device requires plenty time to process data and hence negatively affect to human cognitive interaction.

Based on restrictions from all previous works listed above, we would like to investigate the identification mechanism based on gait signal mined from mobile sensors (e.g. accelerometer, magnetometer) that help to improve two disadvantages: (1) First, the identification mechanism must be more effective than those of previous works and it could operate in a more realistic environment. We improve processing steps such as segmentation, feature extraction from our previous work [8] to increase the identification accuracy. (2) Second, the time delay for performing identification mechanism should be minimized to optimize the human cognitive interaction. Hence instead of performing identification process directly with a limited computing resource, mobile device will automatically transfer collected data to be processed by nearby powerful computers. This requires us to construct a VM-based cloudlet infrastructure which could support diverse applications without our setting up specific software environment on each computer. The rest of this paper is organized into 4 sections. Section 12.2 presents state of the art gait identification techniques using sensors including all approaches and restrictions. Section 12.3 presents our proposed method to improve the effectiveness of the identification mechanism and the VM-based cloudlet infrastructure which could support mobile phone to perform identification. Section 12.4 summaries result from our experiment. Finally, discussion will be presented in Sect. 12.5.

12.2 State of the Art

In 2005, Ailisto et al. were the first to propose the gait identification using wearable accelerometer [7] and this area was further expanded by Gafurov et al. [6]. In general, sensors are attached to various positions on human body to record locomotion signal. Sensors used for gait identification are diverse including gyroscope, rotation sensor but acceleration sensor (or accelerometer) is most used. Gait of individual is recognized using (1) Template Matching or (2) Machine Learning. In (1), the acquired signal is preprocessed and then split into segments. Best segments are considered as typical templates which represent to that person. Distance metrics such as Dynamic Time Warping (DTW), Euclidean distance, autocorrelation were used for estimating the similarity score to extract template in training phase or matching in testing phase [2, 5, 7, 8, 10].

Second method is the most widely used to identification issues. In this approach, acquired gait signal was analyzed to extract features in some domains such as time domain, frequency domain or wavelet domain. Extracted feature vectors were then classified using supervised classifiers like HMM, SVM, etc. [3, 4, 8, 9].

In early stages, most of works used standalone sensors (SSs) have been implemented with a variety of success rate, they still have some restrictions. For example, SSs is relatively expensive, hard to attach to human body due to its size and the interface of some special sensors needs to be developed separately. Gait identification has been initially experimented on mobile sensors during recent years [8, 9]. In comparison to SS, mobile sensors are designed to be cheaper, simpler and as a result, the quality is not rather high as SSs. Derawi et al. [10] pointed up that impact by redid Holien's work [11] using mobile acceleration sensor instead of SS and achieved EER of 20.1 % compared to 12.9 %.

In summary, gait identification using both SSs and MSs achieved significant potential result. However, there still exist constraints that could bring about difficulties when apply in practice. In all of works, researchers assumed that there is no sensor installation (in both orientation and placement) error. Sensors were tightened impractically to the equipment such as suite, shoes, or human body. Moreover, we found that in all previous works, gait identification tasks are assigned to standalone PCs rather than using limited mobile resources. Users could deny setting identification mechanism on their phone if the mechanism operates with an unacceptable speed.

12.3 Method

12.3.1 Improving Gait Identification

Data acquisition
Acceleration data were acquired when user walked naturally. Based on the relationships between gravity, acceleration and motion, we present the output of accelerometer as 3-component vectors

$$A = [A_x, A_y, A_z] \tag{12.1}$$

where A_x , A_y , A_z represent the magnitude of the forces acting on three directions respectively.

In fact, it is impossible to ensure that the mobile device will always be placed exactly by the same orientation and placement all the time. Two errors including misplacement and disorientation could occur simultaneously. From our observation, the misplacement error does not significantly affect to accelerometer's sense. However if mobile device is placed with wrong orientation, accelerometer sensing directions are changed that negatively affect to accelerometer sense. A method to deal with this error is transforming acceleration vector from device coordinate system to a reference (or global) coordinate system using transformation matrix. The transformation matrix is calculated based on yaw pitch roll angles. These angles represent the angle changes between mobile coordinate system with world coordination system. These angles are determined by combination of magnetometer and accelerometer.

Data preprocessing
Orientation calibration
From acquisition step, data is collected with an arbitrary orientation. In this step, we calibrate acceleration data to a fixed coordinate system and eliminate the influence of misplacement problem. Rotation matrix (3 $\times$ 3) R could be calculated by yaw (α) pitch (β) roll (γ) angle.

$$R(\alpha, \beta, \gamma) = \begin{pmatrix} \cos\alpha\cos\beta & \cos\alpha\cos\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{pmatrix} \tag{12.2}$$

Let $A' = [a'_X \, a'_Y \, a'_Z]$ be the raw acceleration signal corresponding to mobile coordinate system. To transform the acceleration signal A' to acceleration signal $A = [a_x, a_y, a_z]$ corresponding to world coordinate system, we multiply the rotation matrix N with A' as $A = A'R$.

Data segmentation
First, we apply linear interpolation to align acquired signal to a fixed sampling rate of 32 Hz. The wavelet decomposition technique (Db6 level 2) is also adapted to eliminate noise. Data segmentation is the most important preprocessing step that

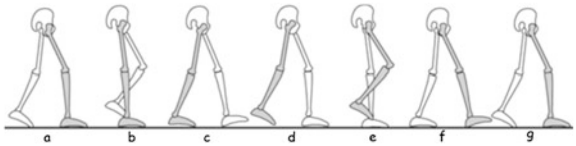

Fig. 12.1 Illustration of a gait cycle

could directly affect to the effectiveness of the system. Hence, we would like to investigate deeply this step to achieve best separated segments. Gait identification is based on walking style of individuals. Hence, data should be segmented based on gait cycles (Fig. 12.1) instead of a fixed time length (e.g. 5 or 10 s) as usual.

From our previous work [8], we designed an algorithm to detect these gait cycles. The algorithm is designed to filter noisy peaks based on a threshold calculated by mean and standard deviation combined with a user-defined constant value. However, the threshold is not robust especially in case that subject walks with a soft state. In this case, steps are not display vividly through high-magnitude peaks. In this work, we improve the segmentation capability by applying an additional correlation method on Z-axis signal to estimate the approximate time gap between two consecutive gait cycles.

$$A = \sum_{i=1}^{N=|m|} x_i x_{i+m} \tag{12.3}$$

where A is the autocorrelation coefficient, x_i is the time series data point, x_{i+m} is the time-lagged replication of the time series.

Feature extraction and classification

In this stage, three phases are investigated to be done in extracting the best feature vector that represents unique characteristics of individuals. First, a total of 38 features on both time domain and frequency domain are extracted on 3 axes data signal. Second, we apply feature a subset selection algorithm called SFFS [12] to obtain the best feature set. The feature subset is selected based on the accuracy criterion of learning algorithm. Hence, the number of features in such subset is reduced to 82 % (7 features). Finally, it is classified using Support Vector Machine (SVM) classifier.

12.3.2 VM-Based Cloudlet Infrastructure

Constrained mobile resources

As discussed above, in this paper we aim to handle the limitation of computational capacity when applying gait identification on mobile devices. Mobile computing systems are constrained in important ways, relative to static systems. These constraints are intrinsic to mobility, and are not just artifacts of current technology. First, mobile device is likely to be optimized by its weight, power, size rather than computational power (e.g. CPU, memory). Second, portable devices

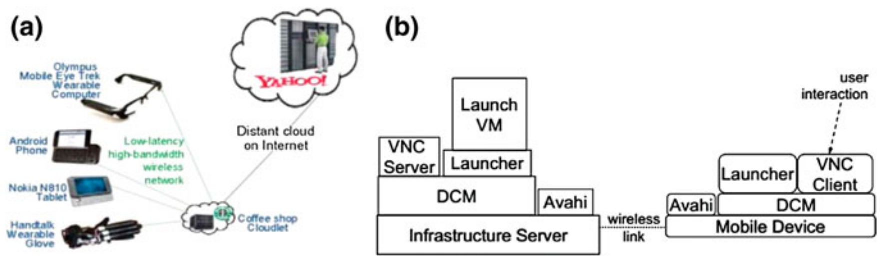

Fig. 12.2 **a** Cloudlet concept with various devices using cloudlet infrastructure. **b** Runtime binding in Kimberley

are more vulnerable to loss or damage, and more difficult to guarantee security on those devices than static devices. Third, battery technology will undoubtedly improve over time, but there is a need to concern for power consumption because of the finite energy source of mobile elements. The state of the art for applications such as image processing, voice processing, gait identification, etc. is near-human in performance and quality today. But they have been done in the lab with ample computing resources, and performing these applications with a poor resource is such a big challenge. For example, it costs $\sim$2 h for learning, and 3 min for prediction when running our identification model on mobile phone.

A solution is to leverage cloud computing. But WAN latency is a fundamental obstacle. Delays induced by transferring large-size data over low bandwidth also hurt latency. For example, to transfer a 10-Mbyte FLV video recorded from built-in camera mobile, it costs 1.5 s with the wireless LAN 802.11 g instead of 40 s with the wireless Internet High-Speed (2 Mbps).

VM-based cloudlet

In cloudlet computing, a mobile device's resource poverty is addressed via a nearby resource-rich cloudlet rather than relying on a distant cloud with high latency (Fig. 12.2). We could achieve real-time interactive response when access to the cloudlet because it has low latency, one hop, and high bandwidth. We also note that the cloudlet infrastructure deployed in a ubiquitous area not only support for a concrete purpose but also various applications. Cloudlet can help to perform widely heavy applications that mobile devices couldn't do with their limited computation resource. We select hardware virtual machine (VM) technology with pre-use customization and post-use cleanup to restore cloudlet infrastructure to its pristine software state after each use.

The Kimberley[1] provides a method accesses one or more application over machines (overlay VMs with small size) from a base machine (base VM with big size). In the Kimberley system, two subsystems execute applications in mobiles on the large displays are called Kimberlize and the Display Control Manager (KCM). Kimberlize is a tool can initialize base VM, execute the installation and execution

[1] Accessed 13-Dec-2012 at http://kimberley.cs.cmu.edu/wiki

scripts (e.g. applications) inside the guestOS and call the overlay VM. KCM is a controller of the transient binding between mobile device and cloudlet. Overlay VMs are assumed that they don't necessarily install the applications before using, but does have good hardware and a strong network connection. Figure 12.2b illustrates the key runtime component of Kimberley. A pair of applications to be implemented in this system is known as the Mobile Launcher and the Display Launcher. Mobile launcher is a tool using to discover service and transfer the data including a VM overlay and the gait signal from mobile to the cloudlet. Display launcher is a tool to broadcast and provide the Kimberley services.

12.4 Experiments

We experimented on data collected from accelerometer in Google Nexus One phone (Table 12.1). A total of 11 volunteers from over 24 year-old participated in data collection. Each volunteer was asked to walk as naturally as possible on the ground floor. They walked for an overall of 12 laps with 36 s on each lap. 5 of 12 lap data were picked randomly for training phase and the other 7 lap data were used to predict. In each lap, they could put the mobile phone freely inside their pocket. The achieved overall accuracy is approximately 98.99 % (Fig. 12.3) which is significantly better than our previous work ($\sim$92.08 %) and other works ($\sim$90–94 %).

By implementing on VM-based cloudlet infrastructure (e.g. Kimberley system in our work), we also reduce the delay time to process signal data expressively. The time delay starting from sending data to nearby computer in cloudlet until receiving returned result only costs $\approx$8 s compared with $\approx$180 s if data is locally processed by mobile resource. The specification of the desktop computer in cloudlet is illustrated in Table 12.2. Table 12.1 compares the performance of identification task between cloudlet infrastructure and local mobile device. The response time is ameliorated significantly so that the interaction between device and user will also be increase. To simulate the Kimberley system, we firstly run kimberlize command to create baseVM and install application to create kimberlize patches. The VMs used in Kimberley are configured using VirtualBox on a Linux host [13]. The kimberlize system will wait until finishing setup current state to create "*.tar.lzma" file. This file is stored directly in mobile storage. After that, we construct several displays on other machines in the same cloudlet infrastructure

Table 12.1 Time performance of gait identification using cloudlet infrastructure versus using local mobile resource

Gait identification using	Response time
Mobile resource	195616 ms
Cloudlet infrastructure	7956 ms

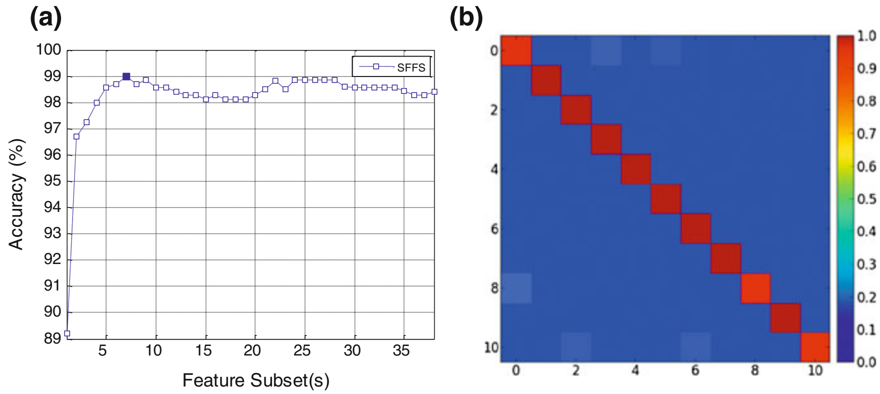

Fig. 12.3 **a** Accuracy of various feature subsets by applying SFFS algorithm. **b** Confusion matrix of the gait identification using the best feature subset selected from (**a**)

Table 12.2 Hardware configuration from our experiment

Mobile part	Local part
HTC Google Nexus One	*Desktop PC*
CPU 1 GHz Scorpion	CPU: Dual Core E2180 2 GHz
Memory 512 Mb Internal	Memory: 3 Gb
Sensors: accelerometer, magnetometer	OS: Linux Ubuntu 8.04—Kernel 2.6.24
Accelerometer: BMA-150 with sampling rate up to 32 Hz, ± 2 g	*Virtual Machine(s)*
	Memory: 1 Gb
OS: Android 2.3.6	OS: Linux Fedora 7.1

using the created baseVM. Then display_launcher command is executed to broadcast Kimberley services.

On mobile side, we launch mobile_launcher to discover and connect to the available display in the operating range. The patch file and gait signal which are stored in mobile storage is then transfer to the display concurrently to create the VM and perform the identification task. Both mobile and server sides use KCM (with Avahi) to broadcast and connect to the services. Cloudlet and a mobile device are communicated over service browsing and publishing using Avahi service discovery mechanism. When identification task is done well, result will be transferred to mobile device and the connection between mobile and cloudlet is closed. A dekimberlize_finished file is created to inform the session is ended, hence the overlayVM will be closed and restore to the previous state.

12.5 Discussion

In this paper, we improved our previous identification mechanism based on gait signal collected from mobile sensor. The mechanism is improved from previous works to achieve the better accuracy. Some restrictions related to laboratory condition were also investigated in order to make the mechanism operate better when deploying in practice. Moreover, the cloudlet infrastructure was also constructed to make the identification process more quickly by assigning tasks to nearby powerful computers, and hence, reducing the time delay and making the mechanism interact better with human cognitive system. However, there are still drawbacks from our experiments. First, the cloudlet infrastructure constructed from Kimberley system seems not to be suitable for our objective that is only focus on reducing the latency. Additionally, the current interface of Kimberley system only supports for USB connectivity which could cause difficulties when deployed in ubiquitous environment. Investigating a novel cloudlet infrastructure that could optimize in transmitting and processing data from mobile is our main future work.

Acknowledgments This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012-035454).

References

1. Breitinger F et al (2010) User survey on phone security and usage, in BIOSIG, vol. 164GI
2. Sun H, Yuao T (2012) Curve aligning approach for gait identification based on a wearable sensor. In: Physiological measurement, vol. 33, no. 6
3. Mondal S, Nandy A, Chakraborty P, Nandi GC (2012) Gait based personal identification system using rotation sensor, In: CIS J 3(3)
4. Hestbek MR, Nickel C, Busch C (2012) Biometric gait identification for mobile devices using wavelet transform and support vector machines. In: IWSSIP 2012
5. Yuexiang L, Xiabo W, Feng Q (2011) Gait identification based on acceleration signals of ankle. Chin J Electr 20(3)
6. Gafurov D, Snekkenes E (2009) Gait identification using wearable motion recording sensors. EURASIP J Adv Signal Proc
7. Ailisto H et al (2005) Identifying people from gait pattern with accelerometers. In: Proceeding of SPIE 5779, biometric technology for human identification II
8. Thang HM, Viet VQ, Thuc ND, Choi D (2012) Gait identification using accelerometer on mobile phone. In: ICCAIS 2012
9. Sprager S, Zazula D (2009) A cumulant-based method for gait identification using accelerometer data with principal component analysis and support vector machine. In: Journal WSEAS transactions on signal processing
10. Derawi MO et al (2010) Unobtrusive user-identification on mobile phones using biometric gait recognition. In: 6th international conference on IIH-MSP
11. Holien K (2008) Gait recognition under non-standard circumstances. Gjøvik University College, Master thesis
12. Pudil P et al (1994) Float search methods in feature selection. In: Patter identification letters 15
13. Satyanarayanan M (1996) Fundamental challenges in mobile computing. In: Proceedings of ACM symposium principles of distributed computing, p 1