# GaitLock: Protect Virtual and Augmented Reality Headsets Using Gait

Yiran Shen, *Member, IEEE*, Hongkai Wen, *Member, IEEE*, Chengwen Luo, Weitao Xu, *Member, IEEE*, Tao Zhang, Wen Hu, *Senior Member, IEEE*, and Daniela Rus, *Fellow, IEEE*

**Abstract**—With the fast penetration of commercial Virtual Reality (VR) and Augmented Reality (AR) systems into our daily life, the security issues of those devices have attracted significant interests from both academia and industry. Modern VR/AR systems typically use head-mounted devices (i.e., headsets) to interact with users, and often store private user data, e.g., social network accounts, online transactions or even payment information. This poses significant security threats, since in practice the headset can be potentially obtained and accessed by unauthenticated parties, e.g., identity thieves, and thus cause catastrophic breach. In this paper, we propose a novel **GaitLock** system, which can reliably authenticate users using their gait signatures. Our system doesn't require extra hardware, e.g., fingerprint sensors or retina scanners, but only uses the on-board inertial measurement units (IMUs) equipped in almost all mainstream VR/AR headsets to authenticate the legitimate users from intruders, by simply asking them to walk a few steps. To achieve that, we propose a new gait recognition model *Dynamic-SRC*, which combines the strength of Dynamic Time Warping (DTW) and Sparse Representation Classifier (SRC), to extract unique gait patterns from the inertial signals during walking. We implement GaitLock on Google Glass (a typical AR headset), and extensive experiments show that GaitLock outperforms the state-of-the-art systems significantly in recognition accuracy ( $>$ 98 percent success in 5 steps), and is able to run in-situ on the resource-constrained VR/AR headsets without incurring high energy cost.

**Index Terms**—Gait recognition, VR/ AR, sparse representation classification, dynamic time warping

## 1 Introduction

Virtual Reality (VR) and Augmented Reality (AR) platforms are booming after recent release of high profile devices, such as HTC Vive, PlayStation VR, Samsung Gear VR, Google Glass, Vuxiz Glasses, and Microsoft HoloLens etc. The race between major industrial players is fierce due to the high profit expectation in the future: according to Goldman Sachs [18], VR/AR devices are predicted to produce over 180 billion dollars (110 billion in hardware and 72 billion in software) in revenue by 2025. Most of the current VR/AR systems uses headsets (head-mounted display equipped with sensors) to interact with users, which have introduced tremendous new applications and services, including home entertainment, cognitive assistance, healthcare, etc. [2], [23], [27], [32], [50].


- *Y. Shen and T. Zhang are with College of Computer Science and Technology, Harbin Engineering University, Harbin, Heilongjiang 150001, China. E-mail: {shenyiran, cstzhang}@hrbeu.edu.cn.*
- *H. Wen is with Department of Computer Science, University of Warwick, Coventry CV4 7AL, United Kingdom. E-mail: hongkai.wen@dcs.warwick.ac.uk.*
- *C. Luo and W. Xu are with College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China. E-mail: chengwen@szu.edu.cn, xuweitao005@gmail.com.*
- *W. Hu is with School of Computer Science and Engineering, University of New South Wales, Sydney 2052, Australia. E-mail: wen.hu@unsw.edu.au.*
- *D. Rus is with Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139. E-mail: rus@csail.mit.edu.*


*Manuscript received 7 May 2017; revised 20 Jan. 2018; accepted 27 Jan. 2018. Date of publication 31 Jan. 2018; date of current version 10 May 2019. (Corresponding author: Yiran Shen.)*
*For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.*
*Digital Object Identifier no. 10.1109/TDSC.2018.2800048*

However, the rise of VR/AR systems also brings substantial risks in security and privacy. Typically the VR/AR headsets are linked to many private online accounts, such as social networks, emails and payment, while the devices may also record our daily routings, activities, and health data. Such information is of significant value, and it is likely to be targeted by malicious parties in the future. To make things worse, in many cases the VR/AR headsets can be easily accessed by other people, e.g., they can be shared among different users, or snapped by thieves just like smartphones, where in those cases the sensitive and private user data is like low-hanging fruits to the potential attackers. The lesson from Internet is that ignoring security at the outset leads to huge pain when the technology becomes ubiquitous. Therefore, in this paper we argue that the ***authentication of users*** of VR/AR devices is a fundamental building block for security, since once the users are authenticated, we can rely on standard methods (e.g., secure communication channel establishment) to achieve the ***integrity*** and ***confidentiality*** of information on VR/AR platforms.

Unfortunately, existing authentication approaches used on desktop or mobile devices are not suitable in the context of VR/AR, since they either require bespoke hardware (e.g., fingerprint sensor is widely adopted as user-friendly and robust authentication mechanism on smartphones and retina scanner is available on few state-of-the-art devices such as Samsung Galaxy Note 7, Microsoft 950 XL, Fujitsu NX F-04G, etc., however, neither of these two sensors can be seen on any popularly accepted commercialised VR/ AR devices such as HTC Vive, Sony VR, Microsoft Hololense and Google Glass), or are not friendly enough for the users

to input (e.g., PIN or passwords) on headsets or cannot be used in natural way due to its special design and use case (e.g., you may have to stand in front of a mirror to apply face recognition). In this paper, we consider the on-board inertial measurement units (IMUs), which have been equipped on most of the VR/AR headsets for head tracking, to authenticate the users seamlessly by checking their gait signatures when walking. This also has many advantages over the existing IMU based gait authentication systems, e.g., they typically need to attach sensors to the wrist [9], shoes [3] or hips [14] of the user, or require the users to make predefined gestures to authenticate, which is prune to imitation attacks [30]. On the other hand, authenticating users using only headsets during normal walking is also very challenging. First, the raw gait features measured on head-mounted devices suffer from gait power attenuation and irregular noises caused by involuntary head motion when transmitted through the body, and thus are not accurate enough for authentication directly. Second, in most cases to better protect the VR/AR systems, instead of offloading the authentication task to the cloud, we require *in-situ processing*, i.e., the sensing and computation involved in the entire authentication process have to be completed on the resource-constrained headsets. Therefore, the gait authentication algorithm has to be extremely lightweight to not jeopardize the normal user experiences of the devices (e.g., latency and battery life).

To address these challenges, this paper proposes a new *GaitLock* system to make high accurate users authentication on resource-constrained VR/AR headsets practical. The proposed system only requires the user to walk normally for few steps (2-5 depends on the users setting), and is then able to authenticate her successfully. The system is based on a novel IMU-based gait authentication model *Dynamic-SRC*, which uses Dynamic Time Warping (DTW) on top of Sparse Representation Classifiers (SRC) to achieve accurate and efficient gait recognition. Concretely, the technical contributions of our paper are:

- We propose GaitLock, a practical user authentication system for VR/AR headsets, which only leverages the widely available on-board IMU sensors to detect intruders, to prevent outlier attacks, and recognize different legitimate users to further support personalization service.
- We propose a new gait recognition model Dynamic-SRC to improve the recognition accuracy of GaitLock. We show that on multiple datasets GaitLock with Dynamic-SRC is up to 20 percent more accurate than existing approaches when recognizing 20 different users, and can also achieve reliable authentication under mimicking attacks (Equal Error Rate (EER) is 2.9 percent).
- We show that by using columns reduction and optimized projections techniques, GaitLock is able to run in-situ on resource-constrained VR/AR headsets. Our evaluation on Google Glass confirms that GaitLock has minimal impact on the energy cost and is able to provide real-time response.

The rest of the paper is organized as follows. We review related work in Section 2, then discuss the application scenarios and threat model we focus on in Section 3, and overview GaitLock in Section 4. The performance of GaitLock is evaluated on collected datasets in Section 5. In Section 6, we implement GaitLock on Google Glass to evaluate its resources consumption on AR headsets. At last, Section 7 concludes the whole paper.

## 2 Related Work

### 2.1 Authentication Systems

Gait recognition was first studied in computer vision community. Vision-based approaches [4], [43] detect and subtract the silhouette of the subject from video recording. Then the scale invariant features [20], [41] of the walking subjects are extracted. At last, different classification methods such as Hidden Markov Model (HMM) [8] are applied to recognize the subject. However, the problem of vision-based gait recognition may be privacy-intrusive because of the image inputs.

The wearable devices are equipped with different sorts of sensors and these sensors can be used to recognize the wearers. These sensors can be bespoke: the authors in [9] studied the bioimpedance information measured from the waist of the subjects via a wearable sensor to continuously recognize the subject wearing it. However, common VR/AR headsets do not feature a bioimpedance sensor. On the contrary, the IMUs are widely embedded in most of mobile/wearable devices and can be used to detect different wearers. For example, Li et al. [25] proposed to use IMUs to monitor the unique head motion to authenticate the wearers when they were listening to a stimulating piece of music.

Gait analysis based on IMUs is popular for authenticating the users of mobile devices, such as authenticating smartphones users [17], [26]. The IMUs attached on smart shoes [3] or hips [10], [14] were also studied to detect the walking pattern of the subject. However the study of gait analysis on head-mounted devices still remains unexplored. Therefore, the above recognition or authentication solutions are not suitable for wearable head-mounted devices.

There also have been many studies on authentication systems using other approaches in the literature [15], [19], [24], [42] which are based on the touching activities on the screen of smartphones. Pan et al. [31] proposed a system that sensed the floor vibration induced by the steps of people to achieve non-intrusive recognition however it required infrastructure deployment. Tian et al. [40] proposed Kinwrite, an authentication system utilizing Kinect to acquire the customized 3D handwriting to distinguish different users. However these approaches are not applicable on head-mounted devices.

### 2.2 Head-Mounted Devices

The research on head-mounted devices is becoming popular. For example [50] proposed a new system iGaze to provide a novel networking mechanism for smart glasses. It understood the interest of connection from the users and connected to a target by a simple gaze. While in [32], the authors studied the problem of tracking the browsing of the users in retail stores to extract key elements in the physical browsing and infer the layout of the stores. The authors in [16] designed Gabriel on Google Glass which provided cognitive services to help users to recognize faces and objects and in [49] and [48] the authors further improved

the performance of cognitive assistance by fusing multimodal sensors. However the authentication systems on head-mounted devices remain vastly unexplored.

### 2.3 Compressive Sensing

Compressive sensing has been popularly used to reduce the resource consumption of the applications on resource constrained systems such as low cost sensors system [21], [28], [47], embedded systems [22], [36], [37], [45] and smartphones [38], [39]. In [47] the authors studied the problem of moisture sensing. The use of random projections and projection matrix optimization were discussed in [36], [38] to reduce the amount of computation in resource constrained systems. The idea of using multiple observations to improve the performance of sparse representation was also investigated in [29], [44] for activities recognition with radio frequencies interference and GPS acquisition.

## 3 Application Scenarios and Threat Model

### 3.1 Multiple Users Authentication and Personalization

Suppose a VR or AR headset is shared by family members for home entertainment or a group of doctors for medical diagnosis assistance. GaitLock is installed in the system to enable the device authenticating the users and personalizing the system accordingly. Specifically, when someone picks up the device, an authentication event will be triggered. Then the screen of the device pops out a message: *Please walk to Authenticate*. Then the unknown subject walks for some steps and the device will determine if he/she is one of the authorized users according to the results from GaitLock. If the subject is recognized as one of the users, the system is unlocked and personalized automatically; otherwise, the access attempt is rejected and another round of authentication process is triggered. When consecutive and multiple authentication attempts fail, the system will be locked in case the adversaries get authenticated occasionally with exhaustive trails. If, though unlikely, the real owner keeps failing for authentication, the system can switch to a cumbersome authentication mechanism, e.g., requests for a preset QR-code to be scanned by the embedded camera of the device to get authenticated.

The proposed authentication system, GaitLock, provides crucial privacy protection service. For examples, as a home entertaining device, it will log on the corresponding social network accounts (Facebook/Twitter) and/or online payment accounts (Paypal/Alipay) when an authorized user is recognized. The information contained in these accounts is quite sensitive and should not be accessed by an intruder or shared with other family members. As a medical diagnosis assistance device, a VR/AR headset may be shared by multiple doctors and it stores and maintains the patient cases for each doctor. Again, these cases should not be leaked to an outlier attacker or accessed by other doctors. Therefore, a proper authentication system should be able to both detect the system intruders and distinguish different authorized users.

### 3.2 Threat Model

The threat model in this paper focuses on two types of adversaries which are, according to the discussion in application scenarios, outlier attackers and inner attackers.

*Outlier Attackers* are the subjects who occasionally obtain the VR/AR headset and are curious about the private information stored inside without the permission from any of the authorized users. These adversaries can be other colleagues in the same hospital of the users, after-hours cleaning staffs or some unknown individuals picking up the lost glasses or thieves snapping it deliberately. *Inner Attackers* are the authorized users of the device who are curious about the information of other users. It is easy for inner attackers to physically obtain the device and conduct attacks.

The "Lunchtime attack" is one of the typical attack examples [13] consistent with this threat model where the adversaries (i.e., colleagues) temporarily obtain the device while the owner is out for lunch. This threat model is typically adopted in authentication systems on mobile devices such as face authentication on smartphones [38], movement pattern authentication on smart glasses [25] and bioimpedance authentication on wrist-worn devices [9] etc. The attackers may perform two sorts of attacks, i.e., *Mimicking Attacks* and *Zero-effort Attacks*. It is possible the attacker is familiar with the walking pattern of the owner or records it beforehand (especially for the inner attackers) so that the attackers can try to mimic the gaits of the owner to get access to the private information stored in the devices. While in zero-effort attacks, the attackers walk for some steps in arbitrary way.

## 4 System Design

In this section, we will provide an overview of the proposed users authentication system, GaitLock on VR/AR headsets. The overall architecture of the system is presented in Fig. 2 and the corresponding sections and details are also labelled for the ease of the readers. The system consists of three main components: 1) offline dictionary building and projection matrix learning, 2) online step cycles extraction and 3) online users authentications.

To make life easier, we first list all the notations used in the proposed algorithm in Table 1.

### 4.1 Step Detection and Interpolation

The step cycles are separated by applying a simple step detection algorithm on the sensor data from accelerometer by finding the local maximums of the step power. However the raw sensor data is corrupted by high frequency noises and irregular head motions as shown in Fig. 3. To reduce the high frequency noises and eliminate the effect of irregular head motions, a bandpass butterworth filter [6] is designed according to the frequency range of walking steps, i.e., the filter order is 4 and passing frequency range is 0.8 Hz to 2 Hz. We choose this very narrow passing bandwidth to make the peak detection easier. Then the local maximums are found to separate the long sequence into step cycles as shown on the lower figure in Fig. 3.

After the step detection, the sensor data of IMUs is separated into short segments of step cycles according to the found maximums. It is noted that the filtered data is only used for finding the segmentation points for the original IMUs data and we feed the segmented original IMUs time series data into gait recognition. After segmentation, we find that most of the step cycles generated by a walking pattern last between 0.4-1.0 s (20-50 samples). This result is in


(a) HMDs (b) IMUs

Fig. 1. Left: Examples of AR/VR headsets in the market; right: IMUs and its axes.

return used to exclude the cycles not produced by normal walking, i.e., the extracted cycles that last less than 0.4 s or over 1.0 s will not be considered.

SRC requires that the signals are of the same length, while the length of the time-series signals varies even when they are all from the normal walking mode. In [1], [5], to apply SRC for activities recognition, zeros are padded at the end of the time series signals until they are in the maximum possible length. However, padding zeros will affect the performance of SRC based gait recognition significantly as the step cycles are not well aligned. In this paper, we apply linear interpolation to approximate the step cycles into the length of 50.

## 4.2 Dynamic-SRC Model

The accurate recognition of GaitLock is based on the new gait recognition model, Dynamic-SRC. Dynamic-SRC takes the sensors data from both accelerometer and gyroscope for gait recognition. It is known that head motions deteriorate IMU measurements. We observe that head motions are significant and mostly occur in yaw direction when walking. The evaluation discussed in Section 5.3 indicates that including gyroscope data from yaw direction will deteriorate the recognition accuracy. To reduce the influence of the head motions, we first exclude the gyroscope sensor data from $z$ axis (as shown in Fig. 1) of the gyroscope which corresponds to the motions in yaw direction of the head. Then we apply the bandpass butterworth filter (order 4, passing frequency range 0.8-2 Hz) on the sensor data obtained from the rest of the sensors axes to filter out the infrequent head motions occurring in other directions. Then, one step cycle can be represented by a data matrix $S = \{s_i\}_{i=1:F}$ where each column $s_i$ is a vector of samples from one sensor axis after interpolation and its length is $l = 50$, and $i$ is the index of the chosen axes. $F = 5$ as it includes all the three axes of accelerometer and the two axes ($x$ and $z$) of gyroscope. $\mathcal{S}_v \in \mathbb{R}^M$ is a vectorized format of $S$ by concatenating the vectors of samples from different axes. Therefore the length of $\mathcal{S}_v$ will be $M = Fl$ (i.e., 250 in our setting).

### 4.2.1 Sparse Representation

To model the gait recognition as a sparse presentation problem, one needs to first build a dictionary $\mathcal{D} \in \mathbb{R}^{M \times N}$ from

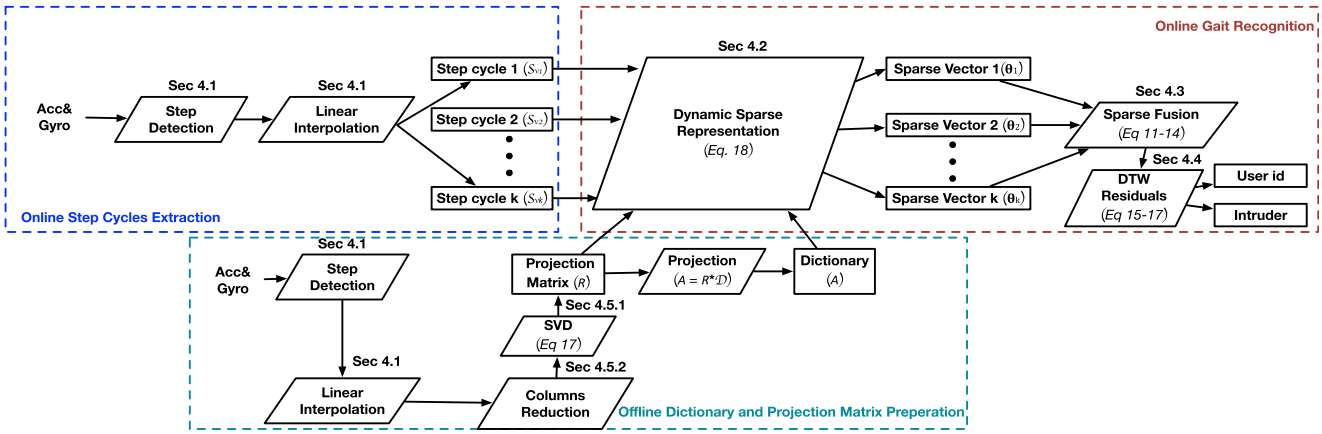



Fig. 2. Flowchart of GaitLock on VR/AR headsets.

TABLE 1
Notations Used the Proposed Algorithm

| | | | |
|---|---|---|---|
| $S$ | data matrix of training step cycle | $\mathcal{S}_v$ | vectorized training step cycle |
| $Y$ | data matrix of testing step cycle | $\mathcal{Y}_v$ | vectorized testing step cycle |
| $M$ | length of $\mathcal{S}_v$ or $\mathcal{Y}_v$ | $l$ | length of each column in $S$ or $Y$ |
| $\mathcal{D}$ | training set | $M, N$ | size of matrix $D$ |
| $P$ | number of classes | $p$ | index of classes |
| $\theta$ | sparse representation coefficients | $\gamma$ | DTW distance between time series |
| $k$ | sparsity of the solution | $\theta_0$ | starting point of the solution $\theta$ |
| $g(\theta)$ | object function of $\ell_1$ optimization | $v_1$ | subgradient of object function |
| $sgn(\cdot)$ | sign function | $\partial_\theta$ | gradient of object function |
| $C_i$ | residual correlation at the $i$th iteration | $d_i$ | step size of the $i_{th}$ iteration |
| $I$ | sparse support set | $\lambda$ | tune parameter in $\ell_1$ optimization |
| $\beta_i^-$, $\beta_i^+$ | solutions at the $i$th iteration | $d_i$ | step size of the $i_{th}$ iteration |
| $\eta$ | set of consecutive steps | $\Theta$ | set of sparse representations of consecutive steps |
| $w_i$ | weight allocated to the $i$th step cycle | $R$ | optimized projection matrix |
| $U, V$ | unitary matrices from SVD | $\sum$ | matrix of singular values |
| $m$ | number of rows of $R$ | $n$ | number of columns of $R$ |

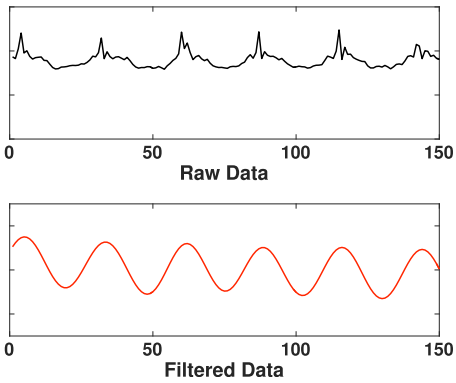



Fig. 3. The time series data and noise reduction.

the training set. Suppose we have $P$ different subjects in the training set and each of the subject contributes a sub-dictionary $\mathcal{D}_p$. The sub-dictionary $\mathcal{D}_p$ contains multiple interpolated step cycles $\mathcal{S}_v$ from the $p$th subject. Then an $M \times N$ dictionary $\mathcal{D}$ from the training set is formed and $N$ is the total number of interpolated step cycles in the training set.

Let $Y \in \mathbb{R}^{l \times F}$ represent the interpolated data matrix of step cycle extracted from the test phase with the same formation of $S$ and $\mathcal{Y}_v \in \mathbb{R}^M$ is the vectorized format. The sparse representation of $\mathcal{Y}_v$ under dictionary $\mathcal{D}$ can be obtained by solving the following $\ell_1$ optimization problem,

$$\min_{\theta} \frac{1}{2} ||\mathcal{Y}_v - \mathcal{D}\theta||_2^2 + \lambda ||\theta||_1, \tag{1}$$

where $\theta \in \mathbb{R}^N$ is the sparse representation of the test step cycle under dictionary $\mathcal{D}$. The $\ell_1$ norm in the object function accounts for the sparseness of the representation while the $\ell_2$ norm accounts for the recovery accuracy. One of the assumptions behind $\ell_1$ optimization is that the test step cycle can be linearly represented by the dictionary $\mathcal{D}$ as $\mathcal{Y}_v = \mathcal{D}\theta$.

However, the interpolated step cycles in dictionary $\mathcal{D}$ are the approximation of the shape of original time-series signals and we cannot guarantee perfect alignment. The optimization result of the objective function in Eq. (1) may lead to non-optimal recognition accuracy. To deal with the problem of poor alignment, we introduce *Dynamic Time Warping Distance* into $\ell_1$ optimization to improve the recognition accuracy.

### 4.2.2 Dynamic Time Warping Distance

Dynamic Time Warping distance is an alignment algorithm developed for matching the time series signals and it is popularly used in speech recognition [35]. DTW distance is applied to measure the similarity of the time series signals with varying speed by warping the time axis iteratively until the optimal non-linear mapping is found. Fig. 4 demonstrates the non-linear mapping path of two time series signals $X = \{x_1, x_2, x_3, \ldots, x_N\}$ and $Y = \{y_1, y_2, y_3, \ldots, y_M\}$. The indices of the $X$ and $Y$ are presented by the indices of rows and columns in the grid. Each cell in the grid represents the measure of the difference between the corresponding elements in $X$ and $Y$. The grid cells with red dots consist of the optimal warping path and the accumulated square distance between the chosen pairs through the path

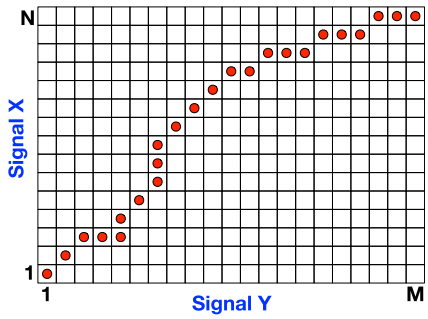



Fig. 4. The matching path of the two time-series signals via dynamic time warping.

is used as the measure of the similarity between the two signals. In mathematics, the accumulated square distance and warping path can be calculated by,

$$\gamma(i,j) = d(x_i, y_j) + \min\{\gamma(i-1,j), \gamma(i,j-1), \gamma(i-1,j-1)\}, \tag{2}$$

where $\gamma(i,j)$ is the accumulated distance until cell $\{i,j\}$. $d(x_i, y_j)$ is the square difference of $x_i$ and $y_j$. DTW finds the warping path that minimizes the accumulated distance between $X$ and $Y$.

As discussed in [34], the computational complexity of standard implementation of DTW is $O(n^2)$. However the fast implementation of DTW will achieve the computational complexity $O(n)$.

### 4.2.3 Incoporating DTW Distance in SRC

Considering the fact that DTW distance is a preferable measure of the distance between time-series signals, the new constraint is added to guarantee the low DTW distance between the test step cycle vector $\mathcal{Y}_v$ and the estimation $\mathcal{D}\theta$ in the $\ell_1$ optimization problem.

*$\ell_1$-Homotopy.* We choose $\ell_1$-Homotopy [12] to solve the optimization problem because it is computationally efficient and has been used in many SRC approaches successfully [38]. The computational complexity of $\ell_1$-Homotopy [12] is $O(k^3 + kMN)$, where $k$ is the sparsity of the solution, $M$ is the number of measurements and $N$ is the dimension of the solution ($k \ll N$, $M \ll N$) which are equal to the number of rows and columns in dictionary $\mathcal{D}$ respectively. According to the discussion in [12], an interior-point based method in general starts from a dense solution and iteratively sparsifies the solution. Different from the general purpose methods, a Homotopy-based solver starts from $\theta_0 = 0$ and iteratively adds and removes non-zeros from the active solution set. As the solution is supposed to be sparse, the Homotopy-based solver is more favorable in terms of efficiency.

Let the object function $g(\theta) = \frac{1}{2}||\mathcal{Y}_v - \mathcal{D}\theta||_2^2 + \lambda||\theta||_1$ ($g(\theta) \in \mathbb{R}^M$). The gradient of the object function $g(\theta)$ will be,

$$\partial_\theta = \lambda \partial ||\theta||_1 - \mathcal{D}^T(\mathcal{Y}_v - \mathcal{D}\theta), \tag{3}$$

where $v = \partial||\theta||_1$ is the subgradient,

$$v_i = \begin{cases} sgn(\theta(i)) & \text{if } \theta(i) \neq 0 \\ [-1, 1] & \text{if } \theta(i) = 0, \end{cases} \tag{4}$$

where $sgn(\cdot)$ is the $sign$ function and $\theta(i) \in \mathbb{R}$ is the $i$th element in $\theta$.

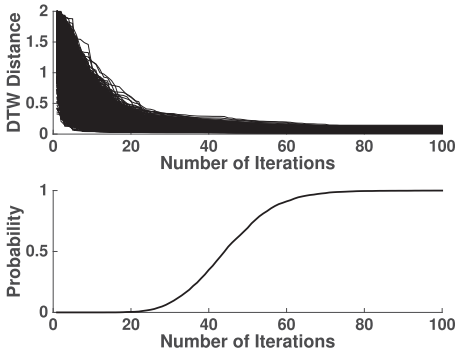



Fig. 5. The convergence of DTW distance.

The Homotopy algorithm finds the optimal solution of $\theta$ that maintains the gradient of object function $g(\theta)$ equal to zero. Specifically, Homotopy starts from an initial solution $\theta_0 = 0$ and iteratively computes the updated solution $\theta_i$. Let the *residual correlation* at the $i$th iteration denoted by $C_i \in \mathbb{R}$ and $C_i = \mathcal{D}^T(\mathcal{Y}_v - \mathcal{D}\theta_i)$ and the sparse support set as $I \subset \mathbb{Z}^+$. At the $i$th iteration, Homotopy finds the update direction $d_i \in \mathbb{R}^N$ by solving,

$$\mathcal{D}^T\mathcal{D}d_i = sgn(c_i(I)), \tag{5}$$

where only the elements with indices in the current active sparse support can be non-zeros. Then the current solution at the $i^{th}$ iteration is computed by,

$$\theta_i = \theta_{i-1} + \beta_i d_i, \tag{6}$$

where $d_i \in \mathbb{R}$ is the step size to the next solution point which has two different possible solutions

$$\beta_i = \min\{\beta_i^-, \beta_i^+\}, \tag{7}$$

where,

$$\beta_i^- = \min_{j \in I}\{-\theta_i(j)/d_i(j)\}, \tag{8}$$

and,

$$\beta_i^+ = \min_{j \in I_c}\left\{\frac{\lambda - c_i(j)}{1 - D_j^T\mathcal{D}d_i(I)}, \frac{\lambda + c_i(j)}{1 + D_j^T\mathcal{D}d_i(I)}\right\}, \tag{9}$$

where $I_c \in \mathbb{Z}^+$ is the set of indices corresponding to the elements not in active support set $I$.

The support set $I$ is then updated by either adding index $i^+$ (the index associate with $\beta_i^+$) or removing index $i^-$ (the index associate with $\beta_i^-$) according to the result of Eq. (7) .

*DTW Distance as Termination Criterion.* After the current solution $\theta_i$ is obtained and the sparse support $I$ is updated, the termination condition will be checked. The key contribution of dynamic sparse representation is the new termination criterion. There are a few existing termination criteria for quadratic optimization algorithms: such as 1) *Relative Change*: the optimization stops when the relative change, defined as $(\theta_i - \theta_{i-1})/\theta_i$, is lower than the threshold which indicates a stable solution has been achieved; 2) *Recovery Errors*: the optimization stops when the recovery error, defined as $||(\mathcal{Y}_v - \mathcal{D}\theta_i)||$, is lower than the requirement which indicates an accurate reconstruction is achieved; it is adopted by our algorithm (see Eq (10)); 3) *Max Iterations*: the optimization stops when the maximum number of iterations are reached. It is often used with other termination criteria to avoid endless loop or for the efficiency consideration. The choice of the termination criteria is important and application-based. For example, as noises extensively exist in real world signals, the signals are often *Compressible* not *Sparse* in some transform domain. A signal is compressible if its representation in some domain has few dominant coefficients and the rest are close to zero. If the optimization algorithm for sparse signals reconstruction does not consider the termination criteria, the feasible solution will not be found as the algorithm intends to find a perfect reconstruction which cannot be achieved. Therefore, the *Recovery Errors* or *Relative Change* with *Max Iteration* can be chosen as the termination criterion for this case.

For the case of gait recognition, DTW distance is known as a favorable measure of the similarity of the time-series signals. Therefore, we exploit the distance structure of the time-series signals based on DTW to propose a new termination criterion which is used to guarantee low DTW distance is achieved. The new optimization algorithm will stop when

$$DTW(\mathcal{Y}_v, \mathcal{D}\theta_i) < \delta, \tag{10}$$

or the number of iterations is over *Max Iteration*.

The near-optimal solution can be guaranteed when $\delta$ is small enough. $\delta$ is not subject or dataset dependent according to our experience. It can be tuned according to the sensor data from the step cycles of one subject while the resultant tolerance is generally applicable for other subjects and different datasets.

To demonstrate that DTW distance will decrease and converge with the growth of number of iterations, we compute the DTW distance between the test signal $\mathcal{Y}_v$ and the estimation $\mathcal{D}\theta_i$ for each optimization iteration in $\ell_1$-Homotopy based algorithm. Fig. 5 presents the results from over 3,000 test signals and each curve in the upper figure stands for the results from one test signal. The lower figure shows the cumulative probability of the number of iterations when DTW distance converges. From the results we can see almost all of the DTW distance from test signals converge within 80 iterations. We set $\delta = 0.05$ and *Max Iterations* as 80 for our system and only less than 0.5 percent of the optimizations terminate due to exceeding *Max Iterations*. As the evaluation results suggest in Section 5.4, the new algorithm achieves significantly higher recognition accuracy than the SRC with traditional termination criteria.

It is worth noting that DTW distance is generally applicable for various $\ell_1$ optimization algorithms. We choose the $\ell_1$-Homotopy based algorithm due to computational efficiency requirement of the system implementation.

### 4.3 Sparse Fusion

Considering the fact that gait information sensed by head-mounted devices is deteriorated by the indirect measurements and noisy inputs of onboard IMUs, the step cycles can be misclassified. The evaluation results in Section 5 also demonstrate that the recognition accuracy of the proposed method with single step cycle is around 86 percent which might not meet the requirement of the security related purpose. To further improve the recognition accuracy, we propose *Sparse Fusion* which fuses the sparse coefficients

vectors from multiple consecutive step cycles simultaneously to improve the recognition accuracy according to the fact that they must be generated by the same subject. In this paper, the steps are said to be *consecutive* before the device is taken off.

Suppose $\eta = \{\mathcal{Y}_{v1}, \mathcal{Y}_{v2}, \ldots, \mathcal{Y}_{v\mathcal{Z}}\} \subset \mathbb{R}^N$, whose columns are the vectorized step cycles extracted from consecutive steps. $\Theta = \{\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_{\mathcal{Z}}\} \subset \mathbb{R}^N$ are the corresponding sparse coefficients vectors under dictionary $\mathcal{D}$. As the step cycles in $\eta$ are known from the same subject, the sparse representations tend to have non-zero coefficients at the same class in dictionary $\mathcal{D}$ while the noises are assumed to be randomly located. Therefore, we propose to use sparse fusion to combine the weighted sparse representations of the consecutive step cycles to improve the Signal to Noise Ratio (SNR) of the classification model

$$\hat{\theta}^C = \Sigma_{i=1}^{\mathcal{Z}} \omega_i \hat{\theta}_i, \tag{11}$$

where $\omega_i \in \mathbb{R}$ is an adaptive weight assigned to the sparse representation of the $i^{th}$ step cycle in $\eta$ determined by the Sparsity Concentration Index (SCI) [46],

$$SCI(\theta_i) = \frac{P \cdot \max_{j=1}^{P} ||\delta_j(\hat{\theta}_i)||_1 / ||\hat{\theta}_i||_1 - 1}{P - 1}, \tag{12}$$

where $\delta_j(\hat{\theta}_i) \in \mathbb{R}^N$ denotes that all the coefficients in $\hat{\theta}_i$ are set to zeros except those related to class $j$. It measures the quality of the sparse representation. The adaptive weight $\omega_i$ is calculated as,

$$\omega_i = SCI(\hat{\theta}_i) / \Sigma_{j=1}^{P} SCI(\hat{\theta}_j). \tag{13}$$

Then the analogous step cycle corresponding to $\hat{\theta}^C$ can be represented as,

$$\mathcal{Y}_v^C = \Sigma_{i=1}^{\mathcal{Z}} \omega_i \mathcal{Y}_{vi}. \tag{14}$$

### 4.4 Intruders Detection and Multiple Users Authentication

With the knowledge of $\mathcal{Y}_v^C$, $\hat{\theta}^C$ and dictionary $\mathcal{D}$, the final classification decision can be determined by computing the residuals of each class. Different from the method using Euclidean distance in [46], we propose the DTW residual.

The definition of the DTW residual for class $i$ is

$$r_i(\mathcal{Y}_v^C) = DTW(\mathcal{Y}_v^C, \mathcal{D}\delta_i(\hat{\theta}^C)), \tag{15}$$

where $\delta_i(\hat{\theta}^C)$ only contains the coefficients related to class $i$ in the weighted sparse representation vector; all the other coefficients are set to be zeros.

*Intruders detection.* Intruders are determined by checking the classification confidence. The classification confidence is defined the same as in [38],

$$confidence = \left( \frac{1}{K} \sum_{j=1}^{K} r_j(\mathcal{Y}_c) - \min_{j=1,\ldots,K} r_j(\mathcal{Y}_c) \right) / \frac{1}{K} \sum_{j=1}^{K} r_j(\mathcal{Y}_c). \tag{16}$$

The confidence is between 0 to 1. It is equal to 1 if the test signal is perfectly represented by step cycles from only one class in the dictionary; the confidence is equal to 0 if residuals are evenly distributed in all classes.

*Users Recognition.* If the subject is regarded as one of the users after checking the confidence, the user ID is determined by finding the class with the minimal DTW residual,

$$\hat{i} = \arg\min_{i=1,2,\ldots,\mathcal{Z}} r_i(\mathcal{Y}_v^C). \tag{17}$$

The gait recognition method presented above is termed as Dynamic (time warping)-SRC based on the two key theoretical components.

### 4.5 Computational Complexity Reduction

Dynamic-SRC is too prohibitive for realtime authentication on VR/AR headsets due to the high dimensionality of the dictionary. We propose to apply optimized projections and columns reduction to improve the efficiency of Dynamic-SRC model while preserving its high recognition accuracy. The fast version is termed as fast Dynamic-SRC.

#### 4.5.1 Optimized Projections

Inspired by the recent advances in information theory of Compressive Sensing (CS) [7], [11], random projection matrices are used in the original SRC [46] to reduce the dimensionality of the $\ell_1$ optimization for sparse representation estimation. Although random projection matrices can significantly reduce the computation time of SRC and are easy to implement, they lead to large performance variance with different projection matrices generated and are not optimal at all. To address the problem of performance variance, optimized projection matrices are proposed [33], [38]. We apply a similar approach to [33] to produce a deterministic and optimal projection matrix to reduce the dimensionality of Dynamic-SRC while preserving the classification accuracy. The projection matrix is learned from dictionary $\mathcal{D}$ based on Singular Value Decomposition (SVD)

$$\mathcal{D} = U \Sigma V^T, \tag{18}$$

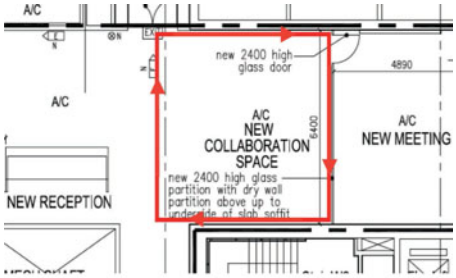
where $U \in \mathbb{R}^{M \times M}$ and $V \in \mathbb{R}^{N \times N}$ are unitary matrices and $\Sigma \in \mathbb{R}^{M \times N}$ a diagonal matrix whose diagonal elements are nonnegative and in decreasing order. The optimized projection matrix $R \in \mathbb{R}^{m \times M}$ is formed by extracting the first $m$ rows of the transpose of the unitary matrix $U$, i.e., the rows corresponding to the first $m$ largest singular values in $\Sigma$. The $\ell_1$ optimization problem is updated by including the projection matrix,

$$\min_{\theta} \frac{1}{2} ||R\mathcal{Y}_v - R\mathcal{D}\theta||_2^2 + \lambda ||\theta||_1. \tag{19}$$

As $m \ll M$ the $\ell_1$ optimization problem is projected into a significantly lower dimensionality.

#### 4.5.2 Columns Reduction

According to the formation of the computational complexity of $\ell_1$-Homotopy, i.e., $O(k^3 + kMN)$, the computation of $\ell_1$ optimization is also proportional to the number of columns in the dictionary $\mathcal{D}$. The step cycles in the same class are highly correlated and lead to intra class redundancy. To reduce the intra class redundancy in the dictionary while preserving the most informative columns, we apply the




(a) Indoor controlled environment


(b) Outdoor uncontrolled environment

Fig. 6. Indoor controlled and outdoor uncontrolled experiments environments.

columns reduction approach proposed in [45] to improve the efficiency of Dynamic-SRC.

According to the evaluation in Sections 5.5 and 6, fast Dynamic-SRC with both optimized projections and columns reduction can preserve the high recognition accuracy while being 25 times faster than Dynamic-SRC with full dictionary.

## 5 Evaluation on Datasets

### 5.1 Goals, Methodology and Evaluation Metrics

The goals of our evaluation are 1) to determine how to reduce the impact of head motions, 2) to evaluate the recognition accuracy of GaitLock and compare it with other state-of-the-art gait recognition methods, 3) to determine the key parameters of GaitLock with fast Dynamic-SRC including the number of projections and number of remaining columns of the dictionary after columns reduction and 4) to evaluate the performance of GaitLock against attacks. *To distinguish different versions of GaitLock, we use Dynamic-SRC or fast Dynamic-SRC to indicate GaitLock with the corresponding models when evaluating the performance of gait recognition.*

In this paper, we only use AR headset, i.e., Google Glass, for the experiments and system implementation because the current VR headsets are only designed for sensing and displaying. However, we envision that, upcoming VR hardwares will be the next big computing platforms and their computational capability should be significantly better than Google Glass due to the contrast of the volumes. We conduct both the *indoor controlled* and *outdoor uncontrolled* experiments and IMUs sensors reading on Google Glass. The details about the datasets collection will be discussed in Section 5.2.

Dynamic-SRC is compared with six different gait recognition methods including Dynamic Time Warping with Nearest Neighborhood (DTW+NN) [10], Time-Delay Embeddings with Template Matching (TDE+TM) [14], Nearest Neighborhood (NN) and three variances of SRC approaches including SRC with zero padding, sparse fusion and majority voting respectively. At last, we evaluate the performance of GaitLock against zero-effort and mimicking attacks.

As discussed in Section 3, a VR/AR device can be shared by multiple users (e.g., family members), GaitLock should be able to both authenticate and differentiate different legitimate users. We use *Recognition Accuracy* to evaluate the performance of GaitLock on differentiating different legitimate users (i.e., identification). Recognition accuracy is the percentage of the correct recognition tests over the total number of recognition tests. The reported recognition accuracy presented in this section is generated from averaging the results from 4-folds cross validation. The performance of GaitLock on authenticating legitimate users is evaluated by multiple metrics: *False Rejection Rate (FRR)*, *False Acceptance Rate (FAR)* and *Equal Error Rate*. FAR is the measure of probability that the authentication system incorrectly accepts the access request by an intruder. FAR is computed as the ratio of the number of false acceptances over total number of access attempts. FRR is a measure of the probability that the authentication system incorrectly rejects the access attempts from real users. FRR is computed as the ratio of the number of false rejections over the total number of access attempts. There is a trade-off between FAR and FRR, i.e., a low FAR may lead to a high FRR or vice versa. EER signifies an equal trade-off between FRR and FAR, meaning that EER is the set of all the points at which FAR $=$ FRR. In short, EER is equal to FAR or FRR when FRR $=$ FAR.

### 5.2 Indoor Dataset Collection

As no public dataset is available for gait analysis with VR/AR headsets, we conduct both controlled indoor and uncontrolled outdoor experiments to collect the suitable datasets.[1] We recruit participants by sending advertising emails within Singapore-MIT Alliance for Research Technology (SMART) and paid incentives to the volunteers. 20 subjects are recruited to collect the datasets. Considering the application scenario described in Section 3 in which the VR/AR headset is shared by family members, where the number of legitimate users is normally not over 5. Therefore, 20 subjects are reasonable to evaluate our system. The recruited group of people age from 20 to 45 and include 8 females and 12 males. The height of people in the group is between 161 cm to 183 cm and the average is 172 cm. The BMI of the people in the group vary from 16.6 $\mathrm{kg/m^2}$ to 27.7 $\mathrm{kg/m^2}$ and the average is 22.3 $\mathrm{kg/m^2}$. To collect the dataset from indoor environment, the subjects participate in two days experiments and each day of experiment consists of two data collecting sessions. The second day of experiment is *one week after* the first day. During the data collecting sessions, the subject was asked to wear the Google Glass while walking along the specific route clockwise (or counterclockwise) as shown in Fig. 6a. The data generated by the IMUs (accelerometer and gyroscope) and its corresponding timestamps are recorded. There are over 60 step cycles in each session. The four sessions are labelled as: 1) clockwise walking in day one, 2) counterclockwise walking in day one, 3) clockwise walking in day two and 4) counterclockwise in day two.

1. Ethical approval was granted by Massachusetts Institute of Technology (Reference Number 1502006877).

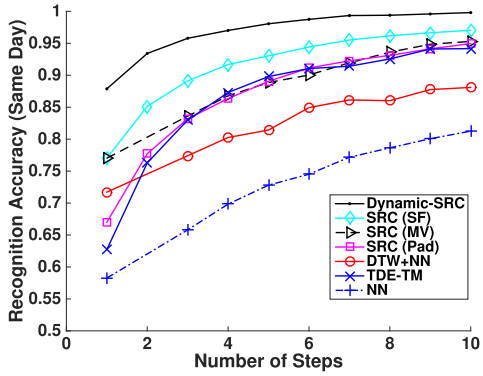



Fig. 7. Accuracy comparison of different gait recognition methods in indoor experiment.

During the evaluation, one session out of four is chosen alternatively as the training set and the rest three sessions as test set. Therefore the training set has approximately over 1,200 steps and test set has over 3,600 steps for each round of evaluation.

### 5.3 Impact of Head Motions

In Section 4.2, we claim that a butterworth bandpass filter should be applied to reduce the impact of irregular head motions and sensor data generated from the $y$-axis of gyroscope should be excluded as head motions in yaw direction are frequent (e.g., turning, talking to the companion or looking around). It is often corrupted significantly and cannot be removed by simple bandpass filter. To verify our hypothesis, we evaluate the performance gain achieved by including the sensor data from each of the three axes of gyroscope. The bandpass filter is applied on the time series data from all axes. From evaluation we find that including the $x$ and $z$ axes of the gyroscope improve the recognition accuracy while the recognition accuracy decreases when including $y$ axis even though the filter has been applied. The gyroscope reading from $y$ axis corresponds to head turning in yaw direction. Therefore, we use the sensor data from accelerometer and $x$ and $z$ axes of gyroscope for the gait recognition.

### 5.4 Comparison with Other Gait Recognition Methods

To satisfy the high recognition accuracy requirement of the security related applications, Dynamic-SRC applies sparse fusion on multiple step cycles to improve the recognition accuracy. To determine whether Dynamic-SRC outperforms the state-of-the-art methods, we also implement DTW+NN [10], TDE+TM [14], Nearest Neighbourhood and the original SRC with different fusion or interpolation choices. In the legends shown in Fig. 7, the original SRC implementation with sparse fusion is represented as SRC (SF). SRC (MV) applies Majority Voting (MV) instead of sparse fusion. SRC (Pad) follows the interpolation method used in gestures or activities recognition in [1], [5] which pads zeros to stretch the signals into the same length.

Fig. 7 presents the recognition accuracy of different methods. The $x$ axis is the number of step cycles used for one recognition and $y$ axis is the recognition accuracy. The recognition accuracy is computed by averaging the results over four rounds of experiments. During each experiment, we choose one session as training set and the rest three sessions as test set. From Fig. 7 we can see that Dynamic-SRC achieves the highest recognition accuracy among the methods implemented and it is 10 percent better than TDE-TM and 20 percent better than DTW+NN at $x = 5$. From the comparison of Dynamic-SRC and SRC (SF), we can find DTW distance improves the performance of SRC for the gait recognition and the proposed approach is up to 10 percent better than original SRC at $x = 1$. SRC (SF) applies sparse fusion to determine the final recognition decision while SRC (MV) uses majority voting. As the results suggest, sparse fusion improves the recognition accuracy by up to 6 percent at $x = 3$. SRC (SF) and SRC (Pad) are different from the interpolation methods. The results indicate padding zeros significantly deteriorates the recognition accuracy and the accuracy difference is up to 10 percent at $x = 1$.

From the results shown in Fig. 7, we can observe that the overall recognition accuracy of Dynamic-SRC increases with the growth of $x$ (the number of steps needed for each recognition). In this paper, we choose a moderate setting as $x = 5$, and the corresponding recognition accuracy can be over 98 percent. However, it is apparent that the authentication system can be more reliable (higher accuracy) when the user would like to pay more efforts on the walking action. Therefore, the number of steps can be a user or application defined parameter.

TABLE 2
Comparison of Spefications of Nexus 5 and Google Glass V2

| | Google Glass V2 | Nexus 5 |
|---|---|---|
| CPU | Dual-core 1 GHz | Quad-core 2.3 GHz |
| RAM | 2 GB | 2 GB |
| Memory | 16 GB | 16 GB |
| Battery | 570 mAh | 2,300 mAh |

### 5.5 Evaluations of Fast Dynamic-SRC

As the discussion suggests in [38], $\ell_1$-Homotopy takes almost $2/3$ of the total computation time therefore is computationally expensive for smartphones. The problem becomes more fierce when it comes to the implementation on smart glasses. Table 2 presents the comparison of Google Glass Explorer Edition V2 and one of the off-the-shelf smartphones: LG Nexus 5 (also released in 2013). According to its specifications, Google Glass is significantly more resource constrained compared with smartphones. Therefore, it is challenging to implement the SRC based approaches on Google Glass. According to the discussion in Section 5.5, we apply optimized projections and columns reduction to improve the efficiency of Dynamic-SRC (i.e., fast Dynamic-SRC) while preserving comparable accuracy. The resource consumption of Dynamic-SRC and its fast version will be evaluated in Section 6. We will only present the performance of accuracy of fast Dynamic-SRC.

#### 5.5.1 *Impact of Number of Projections*

We first apply the optimized projections to reduce the dimensionality of Dynamic-SRC. Dynamic-SRC with random projections are also included as benchmark. The accuracy of random projections is computed from averaging the results from 30 independent trials. Fig. 8 presents the recognition accuracy on different numbers of projections (i.e., the

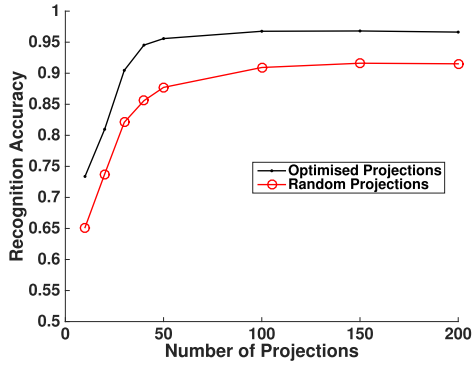



Fig. 8. The recognition accuracy on different number of projections.

number of rows in the projection matrix). The number of steps for each recognition is fixed at 5. From the results we can find that Dynamic-SRC with optimized projections is up to 10 percent more accurate than that with random projections at $x = 4$. Moreover, the accuracy of Dynamic-SRC with optimized projections becomes level when the number of projections is over 50. Therefore, we choose the number of projections as 50 for fast Dynamic-SRC.

#### 5.5.2 Columns Reduction

To determine the minimum number of remaining columns required for each class in the dictionary after columns reduction, we evaluate the recognition accuracy of Dynamic-SRC with different number of remaining columns. We also evaluate the performance of Dynamic-SRC with the columns obtained from uniform subsampling. The step cycles extracted from a walking session are sorted in time sequence. Uniform subsampling evenly picks up the required number of step cycles to form the dictionary. Fig. 9 presents the recognition accuracy of Dynamic-SRC with different number of remaining columns after columns reduction. As the results suggest, the proposed columns reduction approach produces better recognition accuracy than uniform sampling and the performance gain diminishes when the number of remaining columns are over 20. Therefore, only the 20 most informative step cycles are chosen from columns reduction to form the dictionary for fast Dynamic-SRC.

We call the efficient implementation of Dynamic-SRC after optimized projections and columns reduction as fast Dynamic-SRC. The values of the parameters are determined according to the evaluations in this section, i.e., 5 step cycles

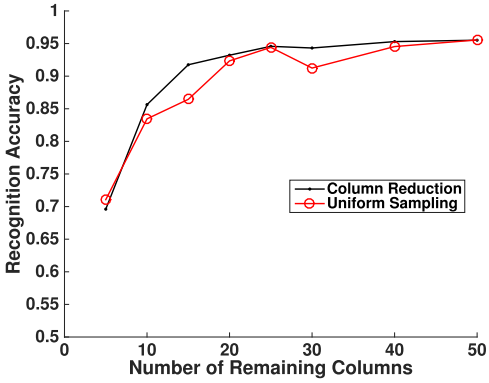



Fig. 9. The recognition accuracy on different number of columns in each class.

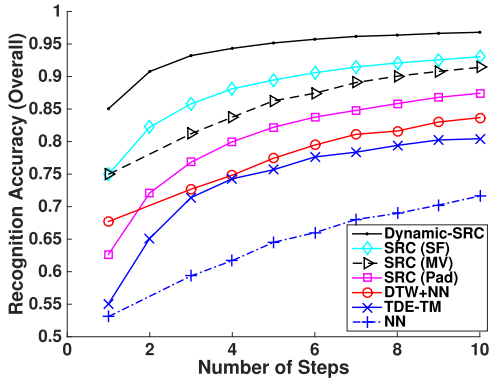



Fig. 10. Accuracy comparison of different gait recognition methods in outdoor experiment.

for each recognition, 50 optimized projections and 20 remaining columns for each subject in the dictionary. As the evaluation suggests in Section 6, fast Dynamic-SRC is over 25 times faster than the original Dynamic-SRC implementation on Google Glass and it only takes less than 900 ms after the required number of steps (i.e., 5 as our setting) are taken according to our evaluation on the computational efficiency in Section 6.

### 5.6 Uncontrolled Outdoor Experiment

To provide more convincing evaluation results, we conduct the outdoor *uncontrolled* experiments which are believed to simulate a significantly more general application case. The same group of people as in the indoor experiments are recruited for the outdoor experiment and the experiment also consists of two sessions. During the first session, the participants were asked to take *arbitrary paths* they like within the outdoor area shown in Fig. 6b. The terrain of the chosen outdoor environment varies including plain, gentle slopes and stairs. The participants were asked to walk freely for 5-10 minutes to collect the IMU sensors' data which accounts for 400-800 step cycles. The second session of experiments is conducted *after one month*. The participants are asked to repeat the experiment and they took significantly *different paths* compared with the previous session. The significant time gap between the two experiment sessions guarantees sufficient variances. We use the step cycles collected from the first session as training set and those from the second session as test set, therefore, the original training set and test set have 20 classes and each set (training or test) has over 12,000 step cycles.

We compare the recognition accuracy of different recognition methods with the parameters determined by the evaluation results on indoor experiments dataset. From the results shown in Fig. 10 we can find Dynamic-SRC still achieves the highest recognition accuracy among the methods implemented and the recognition accuracy is over 95 percent when the number of steps $x = 5$. Comparing with the results from indoor experiments, all gait recognition methods implemented experience performance drop as more dynamics are included in the outdoor experiment. While TDE-TM experiences the most significant accuracy drop: its accuracy drops from the third to the sixth place, which indicates TDE-TM is most vulnerable in the uncontrolled environment.

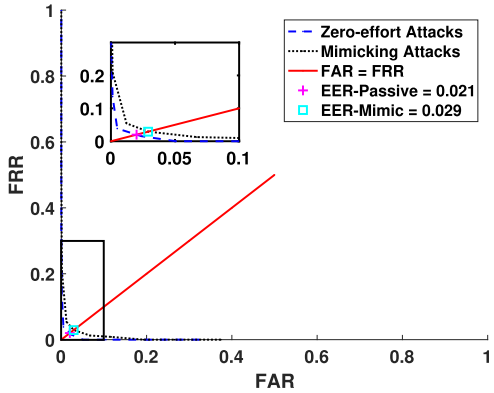



Fig. 11. Verification accuracy in mimicking attack.

### 5.7 Evaluations on Mimicking Attacks

In this section we evaluate the performance of GaitLock on authentication attacks.

We evaluate two types of attacks in this section: *zero-effort attacks* and *mimicking attacks*. It is possible an attacker is familiar with the user or obtains a video in which the user is walking; therefore, he can try to mimic the gait of the owner. In the experiments of mimicking attacks, the 20 recruited subjects are divided into 2 groups: 5 participants are grouped as the users; 15 participants are grouped as attackers. Each of the users wears the Google Glass alternatively and rest of the subjects (including other users) try to mimic the owner's gait. Both of the time-series signals of the users and attackers are recorded for further analyses. To demonstrate the impact of gait mimicking, we also include the *zero-effort attacks* where the attackers try to get authenticated by taking free walks.

By analyzing the recorded time-series signals in the mimicking attacks experiments, the evaluation results of GaitLock against mimicking attacks are presented in Fig. 11 (key region is magnified). The $x$ axis stands for FAR while the $y$ axis stands for FRR. We vary FRR and FAR by changing the threshold for the classification confidence. Higher threshold makes the system more secure (i.e., lower FAR), however, this also brings higher FRR which indicates more steps may be required for succesful authentication due to the failed attemps. We also compute the EER of GaitLock and show the results in the figure. The red straight line consists of all the possible points where FAR is equal to FRR. The crossovers of the red straight line and FRR-FAR curve stands for the location of the EER which is as low as 0.029

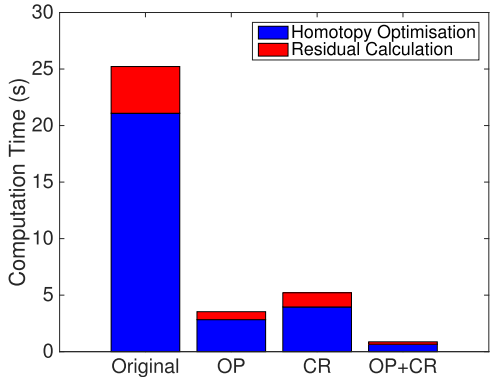



Fig. 12. Comparison of computation time.

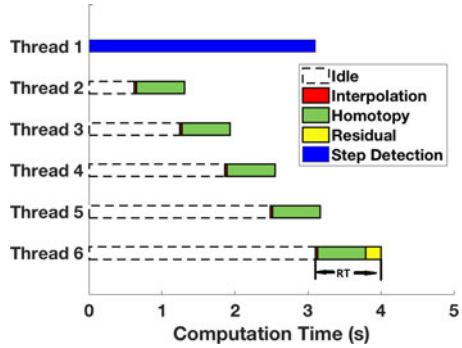



Fig. 13. Demonstration of multi-threads classication.

for mimicking attacks and 0.021 for zero-effort attacks. The results indicates that mimicking user's gait patterns indeed increases the possibility that the attackers get authenticated. The comparable rate of increase is as high as 38 percent. However, the overall EER rate is still quite low (i.e., 0.029) as the benchmark is only 0.021. Moreover, In the real application, the user may change the threshold of classification confidence to satisfy their own needs. For example, a larger threshold makes the system more secure against the attacks while the real users may pay more efforts because it will increase the probability that the real users are detected as the attackers. For example, in our system, we choose the threshold for the confidence level as 0.37 which makes FAR = FRR = 0.029 under mimicking attacks.

## 6 System Implementation and Resources Consumption

In this section, we implement GaitLock in-situ on Google Glass as an authentication system and evaluate its resources consumption. The settings of GaitLock is determined by the evaluation in Section 5.5. Google Glass is popularly used in system research community and they are extremely resource-constrained due to their lightweight and small-size design even compared with other VR/AR devices. Therefore, Google Glass is an ideal choice for benchmarking the in-situ implementation of GaitLock on VR/AR headsets.

*Computational Efficiency.* We first evaluate the improvement of efficiency with columns reduction and optimized projections. The efficiency is represented by the computation time which is obtained from the console of Android studio. Fig. 12 demonstrates the computation time of $\ell_1$-Homotopy and residual calculations with/without columns reduction (RC) or/and optimized projections (OP). The original Dynamic-SRC uses a full dictionary whose number of rows are 250 and number of columns for each class is 60. Fast Dynamic-SRC uses 50 optimized projections and 20 columns for each class. From the results we can find Dynamic-SRC takes up to 25 seconds for each $\ell_1$-Homotopy optimization and residual calculation. However, fast Dynamic-SRC (with both optimized projections and columns reduction) only takes less than 900 ms which is over 25 faster than the original approach.

*Multi-threads Implementation.* To reduce the expected Response Time (RT), we implement GaitLock in multiple threads. RT is defined as the waiting time for the final classification decision after the required number of step cycles are detected. As demonstrated in Fig. 13, six threads are

TABLE 3
Resources Consumption

| | Time | Energy |
|---|---|---|
| Step Detection | 620 ms | 173 mJ |
| Data Interpolation | 29 ms | 8.1 mJ |
| $\ell_1$-Homotopy | 659 ms | 183.9 mJ |
| DTW Residual | 213 ms | 59.4 mJ |
| Total | 4,001 ms | 1,116 mJ |
| Average Response Time | 901 ms | N.A. |

allocated for GaitLock according to the system setting. One of the threads is responsible for the step detection. The rest of the threads are idle before step cycles are received. For each time a new step is detected, the corresponding sensor data of the step cycle is passed to activate a new thread. Threads 2 to 6 interpolate the step cycles into the same length and then compute the sparse coefficients vectors for each step cycle. All of the sparse coefficients vectors are passed to Thread 6. Thread 6 undertakes sparse fusion on all the spare coefficients vectors. After all the sparse coefficients vectors are obtained by dynamic sparse representation, the classification decision is determined by computing DTW residuals on the fused results. Therefore, according to the demonstrated results in Fig. 13, the whole authentication process takes about 4 seconds while the actual RT is about 900 ms (see Table 3) by implementing GaitLock in multi-threads (e.g., 6 threads are used for 5-steps authentication setting). As it takes about 2 to 5 seconds for the next 5 steps, the system produces *realtime* response on Google Glass and will not bring accumulative delay.

*Profiling Resources Consumption.* We then profile the resources consumption of each component of GaitLock. Table 3 presents the detailed results. The computation time and energy consumption are computed by averaging the results from 50 authentication attempts. The energy consumption is profiled by $E = PT$, where $P$ is the average power and $T$ is the runtime of the profiled component. The average power $P = Current \times Voltage$, where the $Current$ and $Voltage$ of the battery is obtained via Android APIs.

From the energy consumption evaluation results shown in Table 3, we can find that each authentication takes about 1.1 J which seems non-negligible if running continuously on VR/AR headsets. However, GaitLock is required only when the VR/AR headsets are turned on or a put-it-on activity is detected because once a successful authentication happens, the user remains authenticated until a take-it-off activity is detected. Meanwhile GaitLock is switched to idle to wait for next triggering activity.

The VR/AR headsets are assumed to be on the same subject's nose before a take-it-off activity is detected. The wearer is regarded as the same user continuously before the take-it-off action is detected once he/she has been successfully authenticated.

The battery capacity of Google Glass Explorer Edition V2 is 2.1 KJ and according to the reviews on Engadget,[2] the battery life of Google Glass only last for 3-5 hours of continuous use. The energy drains even significantly more quickly when recording videos and taking photos. We assume the life span of the Google Glass is 5 hours. Therefore each authentication of GaitLock only accounts for 0.2 percent of the hourly budget (420 J). Considering authentication is not frequent in common usage, GaitLock only has minimal impact on the battery life of the VR/AR headsets.

2. https://www.engadget.com/products/google/glass/

## 7 Conclusion and Future Work

In this paper, we propose a novel authentication and personalization support system, GaitLock, specially designed for VR/AR headsets based on the gait recognition. To address the problem of low recognition accuracy caused by inexpensive but noisy sensor inputs caused by head motions, we design a new gait recognition model, fast Dynamic-SRC. As our evaluation shows, GaitLock is up to 20 percent more accurate than other state of the art implementations on multiple datasets and robust against mimicking attacks. At last, the real world implementation demonstrates that GaitLock can be run in-situ on VR/AR headsets and it has minimal impact on system cost.

However, the current prototype of GaitLock only takes the walking activities into consideration and excludes similar repetitive activities. For example running activities will be ruled out by the constraint on the length of the step cycles in step detection. In the future, we will work on a more comprehensive activities analysis system which detects and distinguishes multimodal activities of the wearers automatically to broaden the usage of the system.

## Acknowledgments

This work is partially supported by National Natural Science Foundation of China under Grant 61702133, 61602319 and U1713212, Natural Science Foundation of Heilongjiang province under grant QC2017069, the Fundamental Research Funds for the Central Universities under Grant HEUCFJ160601, the China Postdoctoral Science Foundation under Grant 166875, Heilongjiang Postdoctoral under grant LBH-Z16042 and Fundamental Research Project in the Science and Technology Plan of Shenzhen JCYJ under Grant 20170302154032530. The authors are grateful to the reviewers and editors for their intensive reviews and insightful comments to improve the quality of this manuscript.

## References

[1] A. Akl and S. Valaee, "Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, & compressive sensing," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2010, pp. 2270–2273.

[2] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, 2010.

[3] S. Bamberg, A. Benbasat, D. Scarborough, D. Krebs, and J. Paradiso, "Gait analysis using a shoe-integrated wireless sensor system," *IEEE Trans. Inform. Technol. Biomed.*, vol. 12 no. 4, pp. 413–423, Jul. 2008.

[4] A. F. Bobick and A. Y. Johnson, "Gait recognition using static, activity-specific parameters," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2001, vol. 1, pp. I–423.

[5] A. Boyali and M. Kavakli, "A robust and fast gesture recognition method for wearable sensing garments," in *Proc. Int. Conf. Adv. Multimedia*, 2012, pp. 142–147.

[6] S. Butterworth, "On the theory of filter amplifiers," *Wireless Engineer*, vol. 7, no. 6, pp. 536–541, 1930.

[7] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.
[8] M.-H. Cheng, M.-F. Ho, and C.-L. Huang, "Gait analysis for human identification through manifold learning and HMM," *Pattern Recognit.*, vol. 41, no. 8, pp. 2541–2553, 2008.
[9] C. Cornelius, R. Peterson, J. Skinner, R. Halter, and D. Kotz, "A wearable system that knows who wears it," in *Proc. Int. Conf. Mobile Syst. Appl. Serv.*, 2014, pp. 55–67.
[10] M. B. Crouse, K. Chen, and H. Kung, "Gait recognition using encodings with flexible similarity measures," in *Proc. 11th Int. Conf. Autonomic Comput.*, 2014.
[11] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
[12] D. L. Donoho and Y. Tsaig, "Fast solution of-norm minimization problems when the solution may be sparse," *IEEE Trans. Inform. Theory*, vol. 54, no. 11, pp. 4789–4812, Nov. 2008.
[13] S. Eberz, K. B. Rasmussen, V. Lenders, and I. Martinovic, "Preventing lunchtime attacks: Fighting insider threats with eye movement biometrics," in *Proc. Netw. Distrib. Syst. Security Symp.*, 2015.
[14] J. Frank, S. Mannor, and D. Precup, "Activity and gait recognition with time-delay embeddings," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 581–1586.
[15] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Trans. Inform. Forensics Security*, vol. 8, no. 1, pp. 136–148, Jan. 2013.
[16] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in *Proc. Int. Conf. Mobile Syst. Appl. Serv.*, 2014, pp. 68–81.
[17] T. Hoang, D. Choi, and T. Nguyen, "Gait authentication on mobile phone using biometric cryptosystem and fuzzy commitment scheme," *Int. J. Inform. Security.*, vol. 14, no. 6, pp. 549–560, Nov. 2015.
[18] G. Sachs, "Virtual & augmented reality: The next big computing platforms," Feb. 2016.
[19] M. Jakobsson, E. Shi, P. Golle, and R. Chow, "Implicit authentication for mobile devices," in *Proc. USENIX Conf. Hot Topics Security*, 2009, pp. 9–9.
[20] L. Lee and W. E. L. Grimson, "Gait analysis for recognition and classification," in *Proc. 5th IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2002, pp. 148–155.
[21] F. Li, J. Luo, G. Shi, and Y. He, "Art: Adaptive frequency-temporal co-existing of zigbee and WiFi," *IEEE Trans. Mobile Comput.*, vol. 16, no. 3, pp. 662–674, Mar. 2017.
[22] F. Li, J. Luo, S. Xin, and Y. He, "Autonomous deployment of wireless sensor networks for optimal coverage with directional sensing model," *Comput. Netw.*, vol. 108, pp. 120–132, 2016.
[23] J.-Q. Li, F. R. Yu, G. Deng, C. Luo, Z. Ming, and Q. Yan, "Industrial internet: A survey on the enabling technologies, applications, and challenges," *IEEE Commun. Surv. Tutorials*, vol. 19, no. 3, pp. 1504–1526, Jul.–Sep. 2017.
[24] L. Li, X. Zhao, and G. Xue, "Unobservable re-authentication for smartphones," in *Network and Distributed System Security Symposium*. Reston, VA, USA: The Internet Society, 2013.
[25] S. Li, A. Ashok, Y. Zhang, C. Xu, J. Lindqvist, and M. Gruteser, "Whose move is it anyway? authenticating smart wearable devices using unique head movement patterns," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2016, pp. 1–9.
[26] H. Lu, J. Huang, T. Saha, and L. Nachman, "Unobtrusive gait verification for mobile phones," in *Proc. ACM Int. Symp. Wearable Comput.*, 2014, pp. 91–98.
[27] C. Luo, H. Hong, L. Cheng, M. C. Chan, J. Li, and Z. Ming, "Accuracy-aware wireless indoor localization: Feasibility and applications," *J. Netw. Comput. Appl.*, vol. 62, pp. 128–136, 2016.
[28] P. Misra, W. Hu, M. Yang, and S. Jha, "Efficient cross-correlation via sparse representation in sensor networks," in *ACM/IEEE International Conference on Information Processing in Sensor Networks*. Beijing, China: ACM, 2012.
[29] P. K. Misra, et al., "Energy efficient GPS acquisition with sparse-GPS," in *Proc. ACM/IEEE Int. Conf. Inform. Process. Sensor Netw.*, 2014, pp. 155–166.
[30] B. B. Mjaaland, "Gait mimicking: Attack resistance testing of gait authentication systems," Master's thesis, Department of Telematics, Faculty of Information Technology, Mathematics and Electrical Engineering, Norwegian University of Science and Technology, Trondheim, Norway, 2009.
[31] S. Pan, N. Wang, Y. Qian, I. Velibeyoglu, H. Y. Noh, and P. Zhang, "Indoor person identification through footstep induced structural vibration," in *Proc. 16th Int. Workshop Mobile Comput. Syst. Appl.*, 2015, pp. 81–86.
[32] S. Rallapalli, A. Ganesan, K. Chintalapudi, V. N. Padmanabhan, and L. Qiu, "Enabling physical analytics in retail stores using smart glasses," in *Proc. Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 115–126.
[33] R. Rana, M. Yang, T. Wark, C. Chou, and W. Hu, "{SimpleTrack}: Adaptive trajectory compression with deterministic projection matrix for mobile sensor networks," *IEEE Sensors J.*, vol. 15, no. 1, pp. 365–373, Jan. 2014.
[34] C. A. Ratanamahatana and E. Keogh, "Everything you know about dynamic time warping is wrong," in *Proc. 3rd Workshop Mining Temporal Sequential Data*, 2004, pp. 22–25.
[35] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 26, no. 1, pp. 43–49, Feb. 1978.
[36] Y. Shen, W. Hu, J. Liu, M. Yang, B. Wei, and C. T. Chou, "Efficient background subtraction for real-time tracking in embedded camera networks," in *Proc. ACM Conf. Embedded Netw. Sensor Syst.*, 2012, pp. 295–308.
[37] Y. Shen, et al., "Real-time and robust compressive background subtraction for embedded camera networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 2, pp. 406–418, Feb. 2016.
[38] Y. Shen, W. Hu, M. Yang, B. Wei, S. Lucey, and C. T. Chou, "Face recognition on smartphones via optimised sparse representation classification," in *Proc. ACM/IEEE Int. Conf. Inform. Process. Sensor Netw.*, 2014, pp. 237–248.
[39] Y. Shen, M. Yang, B. Wei, C. T. Chou, and W. Hu, "Learn to recognise: Exploring priors of sparse face recognition on smartphones," *IEEE Trans. Mobile Comput.*, vol. 16, no. 6, pp. 1705–1717, Jun. 2017.
[40] J. Tian, C. Qu, W. Xu, and S. Wang, "KinWrite: Handwriting-based authentication using kinect," in *Proc. Netw. Distrib. Syst. Security Symp.*, 2013.
[41] D. K. Wagg and M. S. Nixon, "On automated model-based extraction and analysis of gait," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2004, pp. 11–16.
[42] H. Wang, D. Lymberopoulos, and J. Liu, "Sensor-based user authentication," in *Wireless Sensor Networks*. Berlin, Germany: Springer, 2015, pp. 168–185.
[43] L. Wang, T. Tan, H. Ning, and W. Hu, "Silhouette analysis-based gait recognition for human identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1505–1518, Dec. 2003.
[44] B. Wei, W. Hu, M. Yang, B. Wei, and C. T. Chou, "Radio-based device-free activity recognition with radio frequency interference," in *Proc. ACM/IEEE Int. Conf. Inform. Process. Sensor Netw.*, 2015, pp. 154–165.
[45] B. Wei, M. Yang, Y. Shen, R. Rana, C. T. Chou, and W. Hu, "Real-time classification via sparse representation in acoustic sensor networks," in *Proc. ACM Conf. Embedded Netw. Sensor Syst.*, 2013, Art. no. 21.
[46] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.
[47] X. Wu and M. Liu, "In-situ soil moisture sensing: Measurement scheduling and estimation using compressive sensing," in *Proc. ACM/IEEE Int. Conf. Inform. Process. Sensor Netw.*, 2012, pp. 1–11.
[48] W. Xu, Y. Shen, N. Bergmann, and W. Hu, "Sensor-assisted face recognition system on smart glass via multi-view sparse representation classification," in *Proc. 15th Int. Conf. Inform. Process. Sensor Netw.*, 2016, Art. no. 2.
[49] W. Xu, Y. Shen, N. Bergmann, and W. Hu, "Sensor-assisted multi-view face recognition system on smart glass," *IEEE Trans. Mobile Comput.*, vol. 17, no. 1, pp. 197–210, Jan. 2018.
[50] L. Zhang, et al., "It starts with iGaze: Visual attention driven networking with smart glasses," in *Proc. Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 91–102.

**Yiran Shen** received the BE degree in communication engineering from Shandong University, China, and the PhD degree in computer science and engineering from the University of New South Wales. He published regularly at top-tier conferences and journals like ACM SenSys, the *IEEE/ACM Information Processing in Sensor Networks,* IEEE Ubicomp, IEEE Percom, the *IEEE Transactions on Mobile Computing*, etc. His current research interests include wearable/ mobile computing, wireless sensor networks and applications of compressive sensing. He is a member of the IEEE.

**Hongkai Wen** received the DPhil degree from the University of Oxford, and became a post-doctoral researcher in a joint project between Oxford Computer Science and Robotics Institute. He is an assistant professor in the Department of Computer Science, University of Warwick. Broadly speaking, his research belongs to the area of Cyber-Physical Systems, which use networked smart devices to sense and interactive with the physical world. He is a member of the IEEE.

**Chengwen Luo** received the PhD degree from the School of Computing, National University of Singapore, Singapore. He is currently an assistant professor in the College of Computer Science and Software Engineering, Shenzhen University (SZU), China. Before joining Shenzhen University, he was a postdoctoral researcher in CSE, the University of New South Wales, Australia. He is the author and co-author of several research papers in top venues of mobile computing and WSN such as ACM SenSys, the IEEE/ACM Information Processing in Sensor Networks, etc. His research interests include mobile and pervasive computing, indoor localization, wireless sensor networks, and security aspects of Internet of Things.

**Tao Zhang** received the BS and MEng degrees in automation and software engineering from Northeastern University, China, in 2005 and 2008, respectively, and the PhD degree in computer science from the University of Seoul, South Korea, in Feb. 2013. After that, He spent one year at the Hong Kong Polytechnic University as a postdoctoral research fellow. He is currently an associate professor with Harbin Engineering University. His research interests include mining software maintenance, security and privacy for mobile apps, and recommendation systems.

**Weitao Xu** received the bachelor of engineering and the master of engineering degrees from the School of Information Science and Engineering, Shandong University, Shandong, China, in 2010 and 2013, respectively, and the PhD degree from the School of Information Technology and Electrical Engineering, University of Queensland, Australia. He is a member of the IEEE.

**Wen Hu** is a senior lecturer in the School of Computer Science and Engineering, the University of New South Wales (UNSW). Much of his research career has focused on the novel applications, low power communications, security and compressive sensing in sensor network systems, and Internet of Things (IoT). He is a senior member of the IEEE.

**Daniela Rus** is a professor of Electrical Engineering and Computer Science and director of the Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology. Pior to her appointment as director, she served as associate director of MIT Computer Science and Artificial Intelligence Laboratory from 2008 to 2011, and as the co-director of CSAIL's Center for Robotics from 2005 to 2012. She also leads CSAIL's Distributed Robotics Laboratory. She is the first woman to serve as director of MIT Computer Science and Artificial Intelligence Laboratory, and its predecessors the AI Lab and the Lab for Computer Science. She is a fellow of the IEEE.