SilentSign: Device-free Handwritten Signature Verification through Acoustic Sensing

Mengqi Chen, Jiawei Lin, Yongpan Zou, Rukhsana Ruby and Kaishun Wu
College of Computer Science and Software engineering, Shenzhen University, Shenzhen, China
{chenmengqi2017,linjiawei2017}@email.szu.edu.cn, {yongpan,ruby,wu}@szu.edu.cn

***Abstract*—Signature is one of the most prevailing identity authorization approaches. It is yet inconvenient to use in real life in the sense that a majority of existing signature verification approaches rely on additional digital signing devices. In this paper, we propose a portable device-free signature verification system named SilentSign which makes use of acoustic sensors (*i.e.*, microphone and speaker) embedded in smart devices to enable secure and convenient signature verification service. The basic idea is to leverage acoustic signals to measure the distance variation of the tip of the pen while signing. We carefully design the signal modulation scheme, develop a phase-based distance measurement technique, and train the verification model for high performance and robustness. Compared with conventional digital signing systems, SilentSign allows users to sign more invisibly and conveniently. We conduct extensive experiments involving 35 participants to evaluate SilentSign. Results show that SilentSign can achieve 98.2% AUC and 1.25% EER.**

***Index Terms*—Signature Verification, Acoustic Sensing**

I. Introduction

Handwritten signature verification (or, HSV for short) aims at verifying whether a given signature is genuine or forgery, and claiming consent on some obligations [1]. It shows the considerable significance and wide application in our daily life such as signing important documents and handling banking business. However, due to the shortcomings of existing HSV techniques, handwritten signatures are reported to be forged in a large number of serious fraud cases. A recent report filed by JP Morgan shows that fraud associated with paper checks is ranked first for consecutive years among a variety of payment methods [2]. This indicates the significance of HSV research.

Depending on the acquisition method, HSV systems can be classified into two categories, namely, offline verification and online verification [1]. The former refers to writing one's name on paper materials and checks the signature afterward, which is the most traditional HSV approach. This kind of HSV occurs in cases where paper documents such as contracts, receipts and *etc.* need to be signed. But this lacks the real-time verification process of the signer and makes the signature easier to be forged. With the development of hardware, researchers have proposed some novel HSV technology that makes use of smart devices to accomplish online verification of signers and enhance the security of HSV. The underlying principle is to capture dynamic properties of signing movement, such as the order of strokes, speed and pressure of the pen in order to verify the identity of singers, with the help of smart devices such as digitizer and tablet [3]. The feasibility lies in that a signature is associated with several attributes like form, movement and variation which show unique patterns for different people and can be viewed as a person's identity. As a result, in online HSV systems, apart from the signature, the signing process itself can be also utilized for online verifying the identity of signers, which enhances the security of handwritten signatures.

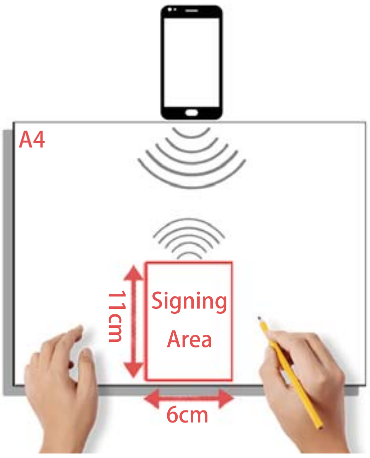


Fig. 1. SilentSign sends inaudible sound signal from speaker to capture the vertical motion variation of a pen's tip during the signing process. Features are extracted from motion variation and used to train a machine learning model to verify the signer.

Within the scope of online HSV, most prior schemes [4]–[7] use a digital signing device such as a digitizer or touchpad, on which users can sign with their signatures with specialized pens. However, these systems require specialized hardware, which makes them not applicable in the cases where users sign on paper materials in daily life. In other words, they can not provide real-time verification service for offline handwritten signature scenarios. The latest work [8] takes advantage of the sensing capability of popular wearable devices, namely, a smartwatch, to capture the wrist movement via inertial sensors for signature verification. Compared with prior works [4]–[7], this novel approach overcomes the shortcomings of the unavailability of hardware. Nevertheless, it still has the following shortcomings. Due to the restriction of computing resources, the smartwatch-based HSV system has to offload collected data to another computing device, for example, a smartphone or a tablet. That is, an HSV system needs to equip with two separate devices. Even with more powerful computing capability, the smartwatch-based HSV technique yet needs another device running applications like electronic banking services owing to its limited screen size. What is more, this method requires a signer to wear a device, which may degrade the user experience.

Consequently, we raise such a question: *can we design an HSV system with only an off-the-shelf device and without the user wearing or touching any additional hardware?* In this paper, we propose SilentSign, an acoustic-based touch-free HSV system that can transform any smart device with acoustic sensors into an online HSV system. SilentSign leverages embedded speaker-microphone pair readily available on commercial smart devices without equipping any additional hardware or making hardware modification. It achieves a fine-grained signature verification objective accurately. The basic idea is to utilize inaudible ultra-sound to capture the vertical trajectory of the pen tip during the signing process as shown in Fig. 1. Then, with image similarity distance as the feature, we train a machine learning classifiers to determine whether the signature trajectory is genuine or forged when an unknown signature comes.

To summarize, we list the following contributions in this work:

- We propose an acoustic-based HSV method that can be easily implemented on readily available smart devices. It can not only supplement real-time signature verification function for scenarios of signing on paper materials but also replace specialized hardware in existing online HSV with a handy device. Compared with similar work [8], it does not require a signer to wear a additional device.
- We design a universal machine learning model for signature verification by combining imaging similarity features (*e.g.*, SSIM, PSNR and Hausdoff distance) that characterize the dynamic pattern of signing trajectories. By such design, SilentSign achieves favorable performance while a new user is enrolled without retraining the model.
- Finally, we conduct extensive experiments and evaluate our system comprehensively. We recruit 35 students and clerks in our University for experiments and collect a total number of 1400 recordings of genuine and forged signatures. The evaluation results show that SilentSign can successfully distinguish genuine and forged handwritten signature at AUC of 98.2% and EER of 2.37%.

The rest of this paper is organized as follows. We outline the related work in Sec. II. We provide the required background information and overview of the architecture in Sec. III. Sec. IV and Sec. V introduce the techniques used in system design and verification model construction. We evaluate the performance of the system in Sec. VI. In Sec. VII and Sec. VIII, we discuss the remaining problems and future work, and conclude this paper respectively.

II. Related Work

A. Handwritten Signature Verification

Relying on the data acquisition type, existing methods for HSV can be divided into two types: offline and online [9]. Offline system uses offline acquisition devices such as a scanner or camera to obtain static images as input data. The verification process is done after the writing process. Current research mainly focuses on the online HSV approach due to its popularity in today's marketplace. Online systems usually rely on dynamic data such as pen pressure, azimuth, altitude and so on. Pen or arm motion data while signing on the paper can be captured by various digitizing tools such as digitizing tablets, special pens and smart wrist [4], [5], [8]. Compared to the aforementioned works, SilentSign novelly uses acoustic signals to track the motion of teh tip of the pen as input data. Moreover, SilentSign has both advantages of online(dynamics data) and offline(Device-free signing on the paper) systems.

B. Biometric Authentication on Mobile Devices

Biometric behavior or biometric authentication on mobile and wearable devices is a popular topic in recent years. Various biometrics such as voice, iris, face and keystrokes, captured by different sensors on portable or wearable devices, have been proved to be used for the purpose of authentication [10]–[12]. Other features such as dental [13] and face [14], heart rate [11] and breath [12] have been used in authentication on mobile or wearable devices as well. On the other hand, due to the unique habits caused by the living environment, behavior biometric is a more trusted feature that can be used for authentication. VibWrite [10] captures the dynamic motion of a finger when a user performs a specific gesture on the touch screen to authenticate its identity. Compared to the aforementioned methods, handwritten signature as authentication feature has been used for a long time in history, its proven uniqueness and application for special occasions is irreplaceable.

C. Acoustic Sensing

Acoustic sensing as a non-contact means of human-computer interaction has broad application scenarios. Due to the range spread and the smaller amount of processed data, sound-based sensing is more advantageous in motion detection or localization, such as gestures by using mobile and wearable devices [15]–[17], indoor localization [18], [19]. FingerIO [15] uses an inaudible OFDM modulated sound frame to locate the moving of finger by detecting the change of two consecutive frames. VSkin [16] leverages the structure-borne and the air-borne sound paths to sense gestures performed on the surface of the smartphone. BeepBeep [18] measures the distance between devices by acoustic ranging. [19] uses a chirp-based ranging sonar achieving the localization error within 1 m. In this paper, SlientSign combines the phase-based and frame-based approach by using Zadoff-Chu coded that has nice auto-correlation properties and the ability to track phase changes. Leading the advantage of a high refresh rate and directly correlate with the movement of the pen tip.

III. System Architecture

A. Considerations

We make the following considerations while designing SilentSign.

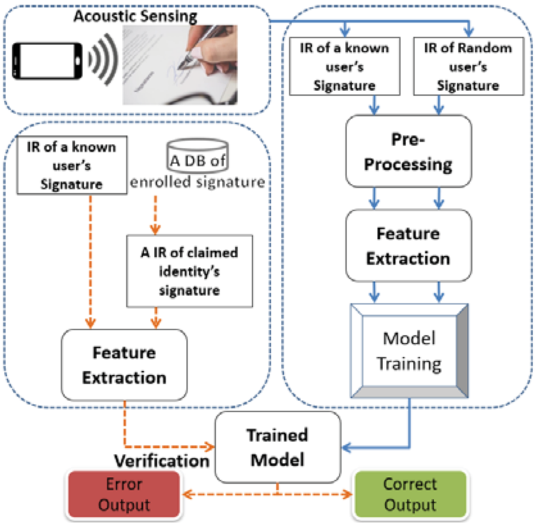

Acoustic Sensing
IR of a known user's Signature
IR of Random user's Signature
IR of a known user's Signature
A DB of enrolled signature
Pre-Processing
Feature Extraction
A IR of claimed identity's signature
Feature Extraction
Model Training
Verification
Trained Model
Error Output
Correct Output

Fig. 2. The system architecture of SilentSign.

- **Sensing Direction:** In traditional online HSV approaches, the dynamics of signatures are captured by a digital signing device with the data modeled as

$$S(t) = [x(t), y(t), p(t)...]^T, t = 0, 1, 2, ..., n$$

  in which, $x(t)$ and $y(t)$ represent coordinates of the pen tip at time t, and $p(t)$... represent other features such as pressure and azimuth. In a typical English signature, the $x(t)$ typically grows linearly with small oscillations on the linear curve, while $y(t)$ changes back and forth between positive and negative values more frequently with more obvious oscillation. Therefore, it is feasible to only consider the vertical movement while ignoring horizontal movement [20].
- **Sensing Accuracy:** The sampling rate of a current commercial digitizer is $75 \sim 200$ Hz with an ideal accuracy about 0.2 mm [21]. Signing is a very delicate action. Using the digitizer, traditional signature verification not only captures the pen tip movement but also pressure and azimuth. Due to the limitation of acoustic sensing, it is impossible to utilize these properties as the features in our system. Therefore, at least we need to make the 1d tracking accuracy as good as the digitizer.

B. Overview

SilentSign system architecture is shown in Fig. 2. The smart device transmits and records inaudible sound with the built-in speaker and microphone. By measuring impulse response and phase changes of received signals, it tracks the movement of a pen tip in the vertical direction. Since the trajectory is a sequence of 2-D vectors, we can regard it as a gray-scale image. We extract traditional image similarity features from two types of trajectory data, genuine and forged, to train an SVM classifier. Moreover, SilentSign consists of two usage stages, namely, signature enrollment and signature verification. In the former stage, the users supply enough number of signatures as the template samples. In the authentication stage, the user just requires to sign his/her signature in the sensing range and the system conducts verification.

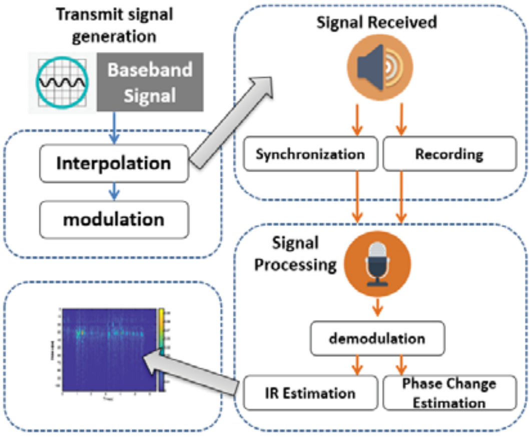


Fig. 3. The overview of a sample acoustic sensing system.

In the authentication phase, the user just requires to sign his/her signature in the sensing range of SilentSign to track the pen movement. SilentSign extracts response and phase change which we mention in the enrollment phase. The features are extracted by comparing input data and stored data and then fed into the trained SVM classifier for final verification.

IV. System Design

A. Acoustic Sensing

There are the following considerations during designing the acoustic sensing part of SilentSign.

1) Current digitizers have a sampling rate of $75 \sim 200$ Hz and a tracking accuracy of 0.2 mm. To achieve better verification accuracy, the performance of our acoustic sensing system should be close.
2) Due to the multipath effect, received echoes are reflected from multiple objects. Thus, we need to differentiate the path corresponding to the moving pen from others.
3) For better user experience, we transmit and record inaudible sound.

To meet these requirements, we design an acoustic sensing system as shows in Fig. 3. It consists of three major components including signal generation, signal reception and distance measurement. The signal generation component is responsible for generating inaudible modulated ultrasound signal. The signal reception component receives reflected sound signals from surrounding objects via microphone, and then synchronizes the sender and receiver. The distance measurement component demodulates the received signals to extract the distance variations between the smartphone and moving pen tip. This component consists of three steps, namely, estimating different impulse responses, estimating the phase change and formatting the impulse response. We shall describe each component in a detail as follows.

B. Transmit Signal Generation

A phase-coded pulse is usually used in radar applications. Giving a pulse, it can be divided into N bit sequence denoted as $S = \{s[1], ..., s[N]\}$ with each bit coded with different phases. Finding a specific code with excellent resolution is criteria due to the unlimited number of possible phase codes. A manageable solution is to find a code with a good autocorrelation function. In this paper, we choose 127 bits Zadoff-Chu coded (ZC sequence), because of its ideal periodic autocorrelation function properties. Furthermore, [16] has proved 127 bits ZC sequence can track moving object with an average movement distance error of 3.59 mm and 3 KHz. These properties are very close to the digitizer.

To generate an adaptive inaudible ZC transmit signal, we first modulate raw 127 bits ZC sequence ($ZC_{127bits}$) into $17 \sim 23$ KHz. Then we apply the frequency domain interpolation on $ZC_{127bits}$ by padding zeros in the middle of the $ZC_{127bits}$ in frequency domain until the length of sequence reach 1024 bits. After this processing, we get interpolated ZC sequence ($ZC_{1024bits}$) which the bandwidth of the result sequence is about 6 KHz at the sampling rate of 48 KHz. Then, we modulated the interpolated ZC sequence into the passband by multiply the real part and imaginary part of $ZC_{1024bits}$ with a carrier. The carrier frequency is 20.25 KHz. Finally, the frequency of transmit signal S_{ZCT} is in the range of $17.29 \sim 23.25$ KHz.

For synchronizing the sender and receiver, we add 24000 zeros followed by $ZC_{1024bits}$ in the very beginning of S_{ZCT}. In the latter part of this section, we will explain how it works. Generated transmit signals can be saved as a WAV file then played by the speaker of the smartphone. The microphone starts recording while the speaker is playing the sound. After receiving the reflected signals, we first use an adaptive energy-based synchronization approach to synchronize the sender and receiver. Then, we demodulate the received signals by down-converting passband signals into baseband ones.

C. Processing of Received Acoustic Signal

Traditional sonar systems can synchronize the sending and recording operations of the signal. After starting the recording operation, the sonar concurrently manages the buffering of the received signal and calculates the distance of the reflected path. Synchronization of the sender and receiver provides a reference for the delay between the sending and receiving time of initial pulse through line-of-sight (LOS). Without synchronization, the delay between initial pulse and first received pulse may not accurately present the time interval of pulse travel through LOS, which will cause deviation to subsequent distance measurement. Due to the compatibility issue of the android operating system, it is difficult to synchronize speaker and microphone.

1) Adaptive Energy-based LOS Detection: To solve this problem, we add 24000 point of zero at the very beginning of S_{ZCT}. 24000 points last 0.5 second that makes sure the recording operation of the microphone before transmitting the pulse. This allows the microphone to receive first pulse completely. Since our acoustic sensing system base on monostatic sonar, speaker and microphone is fixed on the smartphone which means we have already known the length between speaker and microphone. In other word, we know the length of LOS and travel time between sent first pulse and received it. Then, after we found first pulse of LOS, we use it as reference of start time by simply adding fixed delay. Fix delay is based on the distance of speaker and microphone.

To locate the first pulse, SilentSign adopts an adaptive energy-based LOS detection technique to find a precise LOS path. We add $ZC_{1024bits}$ in the following 24000 zero points. Raw 1024 bits ZC sequence has a high auto-correlation gain. Once the recording is started, we perform the cross-correlation function $IR(t) = ZC^*_R(-t) * ZC_{1024bits}(t)$ to obtain impulse response, where $ZC^*_R(-t)$ is the conjugation of received baseband signal. Fig. 4 shows the impulse response of initially received pulse, due to the ideal periodic autocorrelation properties of the ZC sequence. The auto-correlation of the ZC sequence has a low auto-correlation side lobe level, and the first peak is the LOS path.

After applying the cross-correlation function, the next step is to precisely find the position of the LOS peak. For this purpose, we use an adaptive energy-based algorithm to find the rough starting point of the LOS path. We assume that the remaining noise power follows the Gaussian distribution. $\mu(t)$ and $\sigma(t)$ are the average power and its standard deviation at time t. We denote the amplitude of the IR by a discrete series $IR(t)$ and use a sliding window of width W to calculate the average noise power. $\mu(t)$ and $\sigma(t)$ are calculated by

$$\mu(t) = \frac{1}{W}A(t) + (1 - \frac{1}{W})\mu(t-1)$$
$$\sigma(t) = \frac{1}{W}B(t) + (1 - \frac{1}{W})\sigma(t-1)$$

where

$$A(t) = \frac{1}{W}\sum_{k=t}^{W+t} |IR(k)^2|$$
$$B(t) = \sqrt{\frac{1}{W}\sum_{k=t}^{W+t} (|IR(k)|^2 - A(k))^2}$$

$\mu(0) = 0$ and $\sigma(0) = 0$, $A(t)$ is the accumulated power, and $B(t)$ is the overall standard deviation of signals within a sliding window. A rough starting point of IR(t) can be determined if the following relation hold.

$$|S(t)|^2 > \mu(t) + \lambda_1\sigma(t)$$

where λ_1 is a constant which is independent of the noise level. We empirically set W and λ_1 as 1024 points and 18, respectively. As shown in Fig. 4, the red line is a rough starting point, and the peak is within the next 1024 points in the LOS path. Finally, we apply a maximum function to find the exact position of this peak. LOS path is a baseline of the following distance measurement. After adding a fixed delay to the position of the LOS path, we use this position as the starting point of the impulse response.

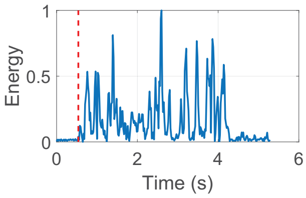


Fig. 4. Adaptive energy-based initial pulse detection.

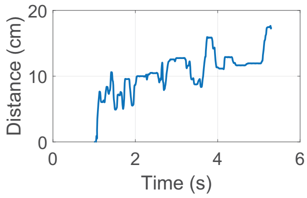


Fig. 5. Distance variation during the signing.

D. Distance Measurement

1) Differential IR Estimations: In modern sonar systems, a sonar transmitter typically sends a known training sequence. Sound signals propagate through the air, meet objects within the detection range, and then reflect back to the receiver in a very short time interval. During this process, the signal is reflected back from multiple different length paths that lead to discrepant time delay and the received signal is a mixture of all paths. To separate different paths at the receiver, we use the cross-correlation function to estimate the Impulse Response (IR). For tracking the moving object, we can locate the changing channel path due to the movement of the object by subtracting the IR between two adjacent time periods. After synchronizing the sender and receiver, we first demodulate the received echoes into baseband signals denoted by $S_{ZCR}(t)$ with a low-pass filter. The Impulse Response (IR) can be estimated by using the cross-correlation function, $IR(t) = S^*_{ZCR}(-t) * S_{ZCT}(t)$. Each peak in the IR estimation indicates one propagation path at the corresponding delay. If the pen tip starts to move, the magnitude of propagation path changes, and then we can achieve these changes by apply subtraction of impulse response between two time's intervals as follows:

$$\Delta IR = IR_{t \sim t+W-1} - IR_{t+W \sim t+2W-1}$$

Furthermore, to save computational cost, we use a standard energy threshold-based algorithm to detect the event when the pen tip starts to move. If the pen tip moves, the position of a maximum point in the ΔIR is the distance between the pen tip and smartphone. However, since the window size is 1024, the frame refreshing rate is about 46.875 Hz, which is below that of digitizers (75 $\sim$ 200 Hz). To increase the refreshing rate, we shall incorporate the estimation of phase change in the following section.

2) Estimation of Phase Change: Once the moving path is detected, we calculate the path coefficient of the moving path to improve the refreshing rate. Finally, by measuring the phase change of the path coefficient, the distance variation of a moving pen tip can be calculated by incorporating the phase change. Path coefficient illustrates how the amplitude and phase of the given path change with time. The formula to compute path coefficient formula as following:

$$h_t[n_i] = \sum_{l=0}^{N_{ZC-1}} S_{ZCR}[t+l] * S^*_{ZCT}[(l-n_i) \mod N_{ZC}]$$

where N_{ZC} is 1024, the length of ZC sequence, n_i is the position of the maximum point in the ΔIR which we will explain in the last section. For computational cost-saving, we only calculate the path coefficient of the moving path. Differential IR Estimations indicate which path is related to moving pen tip, Given the tracked phase change information of this path, the change of distance can be calculated by using the accumulated phase as follows.

$$d_i(t) - d_i(0) = -\frac{\sum_{i=1}^{t} \Delta\theta^i_{i-1}}{2\pi} * \lambda_c$$

where λ_c is the wavelength of sound $\lambda_c = c/f_c$ and

$$\Delta\theta^t_{t-1} = \frac{Q_{h_t}}{I_{h_t}} - \frac{Q_{h_{t-1}}}{I_{h_{t-1}}}$$

We combine low sampling rate IR response estimation with distance change, and then by calculating the maximum point in IR estimation result, the distance variation is a two-dimension array as shown in Fig. 5.

3) Format IR Estimations: Since the initial moving point is uncertain, for the better training, we format IR estimation by the following step. We first scale the value of IR estimation to $0-1$ to build a good model.

$$IR_{norm}(t) = \frac{IR(t) - min(IR)}{max(IR) - min(IR)}$$

Then, we use the following algorithm to remove the beginning point not associated with the detection of the movement and find the moving start point as the center of the array, $IR_{formated}$ is the training sample for the following discussion.

Algorithm 1: IR Estimation Format

```
Input: $IR_{norm}$
Output: $IR_{formated}$
$col = 0; row = 0;$
$center = 0;$
$colSize = ColumnSizeOfIR(IR_{norm}(:)(0));$
while $sum(IR_{norm}(:)(col)) == 0$ do
    $col++$
end
$[value, row] = max(IR_{norm}(:)(col));$
$IR_{formated} = [zeros(colSize - row); IR_{norm}(1 : row + colSize, :)]$
```

V. Authentication Model

A. Feature Extraction

Traditional similarity features such as Structural similarity (SSIM), Peak signal-to-noise ratio (PSNR), Mean squared error (MSE), and Hausdorff distance has been widely used for measuring the image similarity [22], [23]. Therefore, in our system, we calculate the above four features, generate a four-dimension similarity feature vector ($S = \{SSIM, PSNR, MSE, HAUSDORFF\}$) from two scaled and formatted IR response IR_A, IR_B. Depending on whether these two types of IR are generated from the same genuine signature dataset, we label feature vector as *genuine* or *forged*.

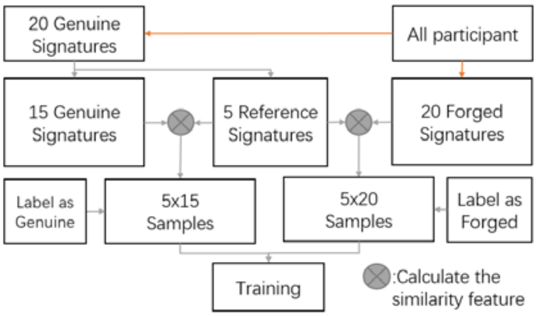


Fig. 6. Model training Design

B. Model Training

1) Training Dataset Enrollment: During the enrollment phase of the training dataset, the user provides a reasonable number of signatures to calculate feature vector F. Based on vector's labels, our training data can be classified into two sets including Genuine Signature VS Forged Signature, and Genuine Signature VS Genuine Signature.

2) Training Phase: For each experimenter, we randomly select 15 genuine signatures as reference signatures denoted by R_i where i represents the index of experimenters. The remaining signature samples are denoted as G_i. His/Her forged signatures denote as F_i. During model training, we first calculate similarity vectors S_G^i between each pair of signatures G_i and R_i, then label them as *genuine*. Meanwhile, we also calculate similarity vectors S_F^i between each pair of signature in G_i and F_i, then label them as *forged*. The training dataset consists of the above two types of samples. After that, we feed the training samples into classifiers to train verification models which decides whether a new input signature is genuine or forged. The training phase is shown in Fig. 6. In traditional signature verification system, a user needs to enroll certain number of genuine signatures as templates and retrain the system. In contrast, our system can be applied to a newly registered user with less retaining. This is because the trained classifier finds thresholds deciding genuine or forged for later used in the verification process. During the system design, we explore four different classification models, including Logistic Regression (LR), Naive Bayes (NB), Random Forest (RF), and Support Vector Machine (SVM), to select an optimal one and achieve better verification performance. We shall detailedly demonstrate the comparison of different models in Sec. VI-B4.

C. Signature Verification

In the verification phase, each new user needs to register his/her signatures in the system first. In this way, the system shall store their signatures as templates in the database and label him/her as *genuine*. When a non-registered person signs, SilentSign will compare the obtained signatures with the stored signatures, calculate similarity vectors, and pass them into a trained classifier to decide whether the signatures are genuine or forged.

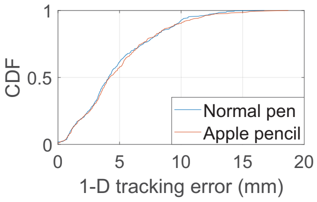


Fig. 7. 1-D tracking errors with normal pen & Apple pencil.

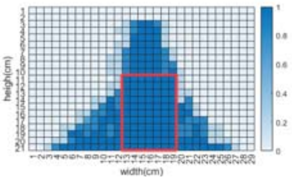


Fig. 8. Acoustic Sensing Range.

VI. Performance Evaluation

A. Acoustic Sensing

1) Tracking accuracy in 1-D: We first evaluate the accuracy of distance estimation with SilentSign implemented on SAMSUNG galaxy note 8. During the evaluation, we attach a ruler of 50 cm on the top of an A4 paper on which we draw a line along with the scale of the ruler to get the ground truth. When we move a pen from a starting position along the line, SilentSign makes use of the speaker and microphone in Note 8 to track the distance change between the pen and smartphone. The ground truth is the length of the line measured by the scale of the ruler. As we use A4 papers in our experiments, the overall testing distance is about 29.7 cm.

In distance estimation, as we use the LOS path as the baseline to measure the following distance, a compensation factor requires to be added to the resultant distance. This is because the initial pulse detected is not the sending time of acoustic signals. Instead, it is the receiving time of the first echo component. Consequently, we need to add the time of flight (TOF) of LOS as a compensation factor, which is essentially a time shift between the sender and receiver. This factor is determined by the distance between a speaker and a microphone. According to our measurements, it is 0.7 cm for SAMSUNG galaxy note 8. We also compare distance estimation performance with different pens. By repeating the above measurement for 400 times, we can obtain the Cumulative Distribution Function (CDF) of distance estimation errors as shown in Fig. 7. As we can see, SilentSign achieves average errors of 4.09 mm and 4.20 mm for normal pen and Apple pencil, respectively. The 90th percentile errors are 9.64 mm and 9.80 mm which are similar to traditional digital signing devices.

2) Tracking Range: Since the speaker and microphone have the directionality property, we evaluate the tracking range of smartphones in this section. The area of A4 paper is 21 cm × 29 cm, we first divide the paper into 609 square blocks (each block has 1 cm length and width). Then we place the smartphone above the centerline of the landscape paper. Second, we draw a circle in each area. If SilentSign acoustic sensing system detects the pen movement within the sensing range, the path corresponding to the pen position will change, the initial value appears in the different impulse responses. We continually check every 609 block and mark it when it has an initial value in the different impulse responses. The experimental result is shown in Fig. 8. Darker area means more sensitive. Since the microphone and speaker of the mobile phone are directional, the closer to speaker and microphone, the narrower the lateral range that can be detected. From our observation, the best sensing region is a 7×11 cm^2 rectangle (the region surrounded by red dashed line in Fig. 8), and the distance between this square and smartphone is 11 cm. Compared to the commercial signature pad (*e.g.*, Wacom STU-300, the signing range is 2.5×9.9 cm^2 [24]), our 7×11 cm^2 signing range is larger and enough for signature verification and avoiding user signing beyond the sensing area. Therefore, we use this region as signing range in the following experiment.

B. *Signature Verification*

1) Data Collection Setup: We recruited subjects to collect genuine and forged signature data. The subjects were asked to sign their names within the signing range on the iPad Pro by using Apple pencil. In their signing process, we place the smartphone above their signature position and turn on the SilentSign app to sense the movement of the pen tips. Although our signing range is relatively large, we did not specifically indicate that the signature must be written in the center of the signing range. As a result, the position of each signature of each participant relative to the mobile phone will randomly move. But the following evaluation results show that this random relative movement did not affect the verification accuracy. Moreover, we conduct our data collection in the rich-noise lab to challenge our system. On the other hand, for collecting forged signatures, we record the screen by using the IOS screen record function. A subject who plays the role of a forger can imitate other subjects with genuine signature by watching recorded signing video. To protect personal privacy, we make a guarantee to each participant that their signature data will not be made public and will only be used in the experiment.

2) Data Collection: In our experiments, we recruit 35 participants including males and females at different ages and with different nationalities from our university. We collect these samples over one month to prove that the performance is time-invariant. The whole data collection experiments include the following two steps.

- *Step 1: collecting Genuine Signatures.* In this step, we collect genuine signatures from 35 subjects. Each of them is required to provide 20 signature samples. Before they sign, we will place the phone above the signature signing range like Fig. 1. In the meanwhile, we turn on the acoustic sensing app to track the pen movement in the vertical direction and record sign trajectory through the screen recording function. Finally, we collect 700 genuine signatures from 35 subjects.
- *Step 2: collecting Forged Signatures.* In this step, we collect the forged signatures of 35 subjects. Each of them is required to imitate 20 samples of a forged signature. We randomly select 5 genuine signatures from other 5 users. Each subject imitates these 5 genuine signatures, and every genuine signature is repeatedly imitated 4 times. Before signing, we play the recorded video of these signatures, and each subject was asked to practice the signatures until before he/she becomes skilled. Finally, we collect 700 forged signatures from 35 subjects. Each subject contains 20 forged signatures, and these signatures are created by other 5 subjects.

3) Signature Verification Setup: To evaluate the performance of genuine signature and random forged signatures, genuine signature and skilled forged signatures is one of the main study topics of any signature verification system. The meaning of a random forged signature is that signature is created without any knowledge. By evaluating this, we can understand whether our system is robust, preventing random signature pass through. Skilled signatures are created with a certain level of training on the genuine signature of the claimed user [8]. As a result, we consider three testing cases to evaluate the verification model of SilentSign. Note that in our dataset, for each subject u, there are 20 genuine signatures denote as G_i and 20 forge signatures denote as F_i.

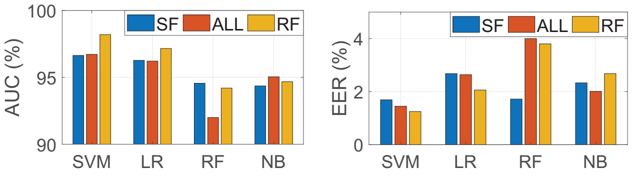
- *Case 1:* distinguishing between genuine signatures and skilled forgeries (denoted as 'SF').
- *Case 2:* distinguishing between genuine signatures and random forgeries (denoted as 'RF').
- *Case 3:* distinguishing between genuine signatures and both types of forgeries (denoted as 'ALL').

All genuine signatures in case 1, case 2 and case 3 are randomly selected from u's G_i, and the 15 forged signatures are randomly selected out of the F_i of u. We select 15 subjects (not including u that we first selected) as a random forger, and then randomly select 1 genuine signature out of his/her G_i for each of those subjects (we have total 15 samples as RF). Finally, we randomly select 15 signatures out of all the signature samples (we have a total 15 samples as ALL).

After labeling signatures, we then calculate the similarity vector between genuine signature and SF, genuine signature and RF, genuine signature and ALL, and fed to a trained classifier for verification. The experiments associated with case 1, case 2 and case 3 have been repeated 50 times. Moreover, we change random seed in each iteration to keep our system generalizable. Final results are the average ones summarized by all iterations. We compare the performance of different classifiers, namely, LR, NB, RF and SVM as mentioned in Sec. V-B2.

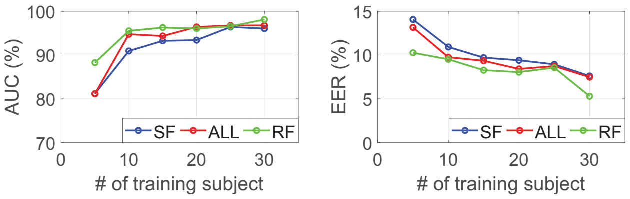
Similar to the works [8], [25], we adopt two main metrics to quantify the performance of SilentSign, namely, area under curve (*i.e.*, AUC) and equal error rate (*i.e.*, EER). AUC is defined as the area under the receiver operating characteristic curve (*i.e.*, ROC). The higher it is, the better the system works. EER is the point on the ROC curve that corresponds to an equal probability of miss-classifying a positive or negative sample. The lower its value is, the better the system performs.

4) Performance of different models: Fig. 9 shows the AUC and EER of four different classifiers. All classifiers perform good, and the SVM model outperform others: AUC = 98.6% and EER = 1.7% in SF, AUC = 96.7% and EER = 1.5% in ALL, AUC = 98.2% and EER = 1.3% in RF. We believe



(a) The performance of different classifiers in terms of AUC (b) Different classifiers performance - EER

Fig. 9. AUC & EER for different classifiers.



(a) AUC (b) EER

Fig. 11. AUC & EER for different number of training Subjects.

TABLE I
AUC & EER FOR DIFFERENCE SIGNATURE COMPLEXITY

	AUC (%)			EER (%)		
	SF	ALL	RF	SF	ALL	RF
Simple	83.1	85.9	85.5	18.9	16.8	16.7
Medium	88.8	91.9	94.4	11.5	6.9	3.1
Complex	93.8	92.4	96.1	3.4	4.2	3.8

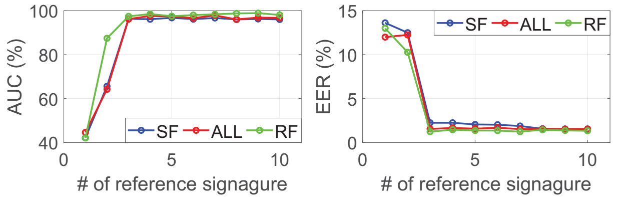


Fig. 10. AUC & EER for different reference signatures.

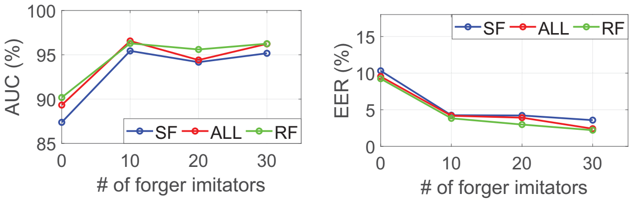
the nature of the features we used is the reason why SVM classifier achieve the best results. The features used are the similarity value between reference and questioned signatures. Genuine signatures are more likely to achieve high similarity value than forged signature. Moreover, the ranking of the SVM classifier for the three tasks is as follows. ALL have the best performance followed by RF, and SF is the worst. These results just satisfy our intuition that skilled forgeries are mostly similar to the genuine signature since the distance variation is almost the same. In the following results, we use the SVM model as the final classifier for signature verification and continually evaluate the verification accuracy of the SVM model.

5) Required number of reference samples: In this section, we evaluate the impact of required number of reference samples. We keep the number of subjects the same and increase the number of reference samples ranging from 1 to 10 for each subject gradually. A trained SVM model is used to classify three tasks we mention in section VI-B3. Fig. 10 shows the AUC and EER for a different amount of reference samples. As the number of reference samples increases, the scores improve rapidly from AUC=42.1% and EER=13.6% using a single reference signature to AUC=97.4% and EER=1.2% using 3 reference signatures. And the best score achieves at task RF, while the number of reference signature is 9, the AUC is 98.9% and EER is 1.3%. From our observation, even if 3 reference signatures seem to be enough, increasing reference signatures amount leading to a robust and secure system.

6) Required number of training subjects: In this section, we evaluate how many training subjects are enough to achieve good performance. For this purpose, we train our models with varying amounts of subjects, starting with 5, adding 5 subjects each time as a training set, the rest are testing set, until the number of testing subjects reaches 30. For example, we have 35 subjects in total. We first randomly select 5 subjects, their genuine and forged signatures as training samples, and the rest of 30 subjects as testing samples. Next, we randomly select 10 subjects as training subjects and 25 as testing subjects. We run the training and testing operations one after another until the number of training subjects reaches 30. In order to ensure that our experiments are not interfered with by abnormal samples, our evaluation has been repeated 25 times. The average AUC and EER are shown in Fig. 11. When the training samples increase, the performance becomes better. The best score achieves at task RF, while the number of training subject is 30, the AUC is 98% and EER is 5.3%.

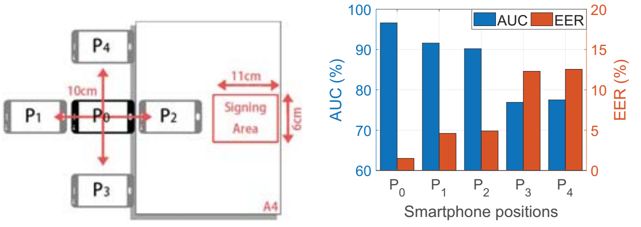
7) Impact of signature complexity: We also evaluate whether the signature complexity influences performance. To do this, we first define three complexity levels of signatures by 'Simple', 'Medium' and 'Complex'. If a signature contains less than 4 letters, over 10 letters, or in between, its complexity is defined to be 'Simple', 'Complex' or 'Medium', respectively. For each level, we select 3 participants with signatures meeting the criteria for verification. The result is showed in Table I. As we can see, with signature complexity gets higher, the AUC increases and EER decreases, indicating the improvement of system performance. It's recommended that using a signature of medium complexity and above.

8) Impact of number of forger imitators in training: SlientSign is intrinsically a non-retrain system for forgers since their signatures are difficult to be obtained in real-world application scenarios. But it is possible to collect data from forger imitators who could be our friends, colleagues, or recruited participants, to improve the authentication ability of our system. Thus, a question should be answered that *how the system performance could be affected by the number of forger imitators in model training*. In response, we randomly divide all the participants into three groups, namely, legitimate users, forger imitators, and real forgers. We train binary classification models with data from the legitimate user and forger imitators and test it with data from the remaining participants (*i.e.*, real forgers). As we can see, when data of more forger imitators are used, the system performance can be enhanced. When the number of forger imitators in the training stage reaches 10, the AUC and EER get around 96% and 4%, respectively.



(a) AUC (b) EER

Fig. 12. AUC & EER under different number of forger imitators included in model training.



(a) The experimental setup to evaluate the impact of smartphone position (b) The AUC& EER when smartphone is placed in dofferent positions

Fig. 13. the performance of SilentSign when the smartphone is placed at different positions.

9) Impact of the smartphone position: We finally evaluate the impact of smartphone position on verification performance. As shown in Fig. 13(a), we move the smartphone from the original position P_0 along four directions for 10 cm and place it at four different positions ($P_1 \sim P_4$). Then we request two participants to perform genuine signatures and forged signatures respectively as described in Sec. VI-B for 10 times at each position. Based on the collected samples, we run the verification process with samples in P_0 as reference signatures and get the results as shown in Fig. 13(b). As we can see, the smartphone position indeed affects system performance. When it is moved away from the original position P_0, the already trained system degrades But for different positions, the impact varied. When the smartphone is moved horizontally (P_1 and P_2), the performance has smaller decrease; while for vertical movements (P_3 and P_4), the performance degradation is more obvious. The underlying reason is vertical movements cause changes in relative orientations between signing activity and the smartphone. This further affects the measurement of vertical movement which is used as a key feature.

VII. Discussion and Future Work

In this part, we mainly discuss the limitations and future work of SilentSign.

A. The impact of relative orientation

Although we have verified with experiments that SilentSign is not sensitive to signing positions within the sensing area, it is to be pointed that this holds true when the device does not move as shown in Sec. VI-B9. Essentially, SilentSign is sensitive to the relative orientation between signing a pen and the device. If the device is moved vertically or rotated relative to the sensing area, the system performance will be negatively affected. This is because the orientation determines the relative movement between signing activity and the device, which in turn affects the echo signals. This is one of the limitations of our system. We envision that this can be improved by extracting orientation-independent features and collecting training data from several orientations in the future work.

B. The impact of lack of forged signatures

As aforementioned, the signatures of real forgers can not be obtained in real-world application scenarios, which causes performance degradation as shown in the evaluation. Although adding data of forger imitators into training can improve the performance, it keeps stable in terms of AUC and EER even when more forger imitators' data are used. To gain further optimization and fulfill the higher requirement in certain scenarios like banking services, it is feasible to design a more advanced verification model by making use of deep neural network which is more powerful to extract deep features. We leave this as one of our future work.

VIII. Conclusion

In this paper, we propose an acoustic sensing-based handwritten signature verification method which can be implemented on handy smart devices such as smartphone and tablets. Compared with the common touchscreen-based HSV system, our method has a lower hardware requirement and can be applied in scenarios of signing on paper materials to supplement real-time signature verification. Our approach is a purely software-based solution and only uses a speaker and microphone which are basic components of most commodity devices. By extracting intrinsic patterns of signing movements, our well-designed system SilentSign can achieve satisfactory signature verification performance in terms of metrics of AUC and EER. Although it still has limitations in practicability and robustness, we believe that this is a promising technology deserving further research.

Acknowledgment

We really appreciate the kind effort of our shepherd, Prof. Petteri Nurmi, for giving precious suggestions to revise and improve the paper. This research was supported in part by the China NSFC Grant (61802264, 61872248, U1736207), Guangdong NSF 2017A030312008, Fok Ying-Tong Education Foundation for Young Teachers in the Higher Education Institutions of China (Grant No. 161064), Shenzhen Science and Technology Foundation (No. JCYJ20180305124807337, No. ZDSYS20190902092853047), GDUPS (2015), and Natural Science Foundation of SZU (No. 860-000002110537). Yongpan Zou is the corresponding author.

References

[1] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Offline handwritten signature verificationliterature review," in *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, 2017, pp. 1–8.

[2] J. Morgan, "2019 afp payments fraud and control survey report," https://www.jpmorgan.com/commercial-banking/insights/2019-afp-payments-fraud-control-survey-report.

[3] A. Pansare and S. Bhatia, “Handwritten signature verification using neural network,” *International Journal of Applied Information Systems*, vol. 1, no. 2, pp. 44–49, 2012.

[4] K. K. Gurrala, “Online signature verification techniques,” Ph.D. dissertation, 2011.

[5] D. Muramatsu and T. Matsumoto, “Effectiveness of pen pressure, azimuth, and altitude features for online signature verification,” in *International Conference on Biometrics*. Springer, 2007, pp. 503–512.

[6] A. Kholmatov and B. Yanikoglu, “Identity authentication using improved online signature verification method,” *Pattern recognition letters*, vol. 26, no. 15, pp. 2400–2408, 2005.

[7] A. A. Jaini, G. Sulong, and A. Rehman, “Improved dynamic time warping (dtw) approach for online signature verification,” *arXiv preprint arXiv:1904.00786*, 2019.

[8] A. Levy, B. Nassi, Y. Elovici, and E. Shmueli, “Handwritten signature verification using wrist-worn devices,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, p. 119, 2018.

[9] J. Fierrez and J. Ortega-Garcia, “On-line signature verification,” in *Handbook of biometrics*. Springer, 2008, pp. 189–209.

[10] J. Liu, C. Wang, Y. Chen, and N. Saxena, “Vibwrite: Towards finger-input authentication on ubiquitous surfaces via physical vibration,” in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 73–87.

[11] C. X. Zhao, T. Wysocki, F. Agrafioti, and D. Hatzinakos, “Securing handheld devices and fingerprint readers with ecg biometrics,” in *2012 IEEE fifth international conference on biometrics: theory, applications and systems (BTAS)*. IEEE, 2012, pp. 150–155.

[12] J. Chauhan, Y. Hu, S. Seneviratne, A. Misra, A. Seneviratne, and Y. Lee, “Breathprint: Breathing acoustics-based user authentication,” in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2017, pp. 278–291.

[13] Y. Zou, M. Zhao, Z. Zhou, J. Lin, M. Li, and K. Wu, “Bilock: User authentication via dental occlusion biometrics,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, p. 152, 2018.

[14] B. Zhou, J. Lohokare, R. Gao, and F. Ye, “Echoprint: Two-factor authentication using acoustics and vision on smartphones,” in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, 2018, pp. 321–336.

[15] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, “Fingerio: Using active sonar for fine-grained finger tracking,” in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 1515–1525.

[16] K. Sun, T. Zhao, W. Wang, and L. Xie, “Vskin: Sensing touch gestures on surfaces of mobile devices using acoustic signals,” in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, 2018, pp. 591–605.

[17] Y. Zou, Q. Yang, R. Ruby, Y. Han, S. Wu, M. Li, and K. Wu, “Echowrite: An acoustic-based finger input system without training,” in *Proceedings of IEEE ICDCS*. IEEE, 2019, pp. 778–787.

[18] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, “Beepbeep: a high accuracy acoustic ranging system using cots mobile devices,” in *Proceedings of the 5th international conference on Embedded networked sensor systems*. ACM, 2007, pp. 1–14.

[19] P. Lazik and A. Rowe, “Indoor pseudo-ranging of mobile devices using ultrasonic chirps,” in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 2012, pp. 99–112.

[20] G. Gupta and A. McCabe, “A review of dynamic handwritten signature verification,” *Department of Computer Science, James Cook University Townsville, Qld*, vol. 4811, 1997.

[21] “Testing the accuracy of pen tablets,” https://neuroscript.net/tablets/reviews_accuracy.php.

[22] E. A. Silva, K. Panetta, and S. S. Agaian, “Quantifying image similarity using measure of enhancement by entropy,” in *Mobile Multimedia/Image Processing for Military and Security Applications 2007*, vol. 6579. International Society for Optics and Photonics, 2007, p. 65790U.

[23] “Hausdorff distance between convex polygons,” http://cgm.cs.mcgill.ca/~godfried/teaching/cg-projects/98/normand/main.html.

[24] “Testing the accuracy of pen tablets,” http://signature.wacom.eu/wp-content/uploads/2014/06/Wacom_factsheet-signature_pad_STU-300-EN.pdf.

[25] A. Fischer, M. Diaz, R. Plamondon, and M. A. Ferrer, “Robust score normalization for dtw-based on-line signature verification,” in *2015 13th international conference on document analysis and recognition (ICDAR)*. IEEE, 2015, pp. 241–245.