Discrimination between Genuine and Cloned Gait Silhouette Videos via Autoencoder-Based Training Data Generation*

Yuki HIROSE†(*), Student Member, Kazuaki NAKAMURA†, Naoko NITTA†, Members, and Noboru BABAGUCHI†, Fellow

SUMMARY Spoofing attacks are one of the biggest concerns for most biometric recognition systems. This will be also the case with silhouette-based gait recognition in the near future. So far, gait recognition has been fortunately out of the scope of spoofing attacks. However, it is becoming a real threat with the rapid growth and spread of deep neural network-based multimedia generation techniques, which will allow attackers to generate a fake video of gait silhouettes resembling a target person’s walking motion. We refer to such computer-generated fake silhouettes as gait silhouette clones (GSCs). To deal with the future threat caused by GSCs, in this paper, we propose a supervised method for discriminating GSCs from genuine gait silhouettes (GGSs) that are observed from actual walking people. For training a good discriminator, it is important to collect training datasets of both GGSs and GSCs which do not differ from each other in any aspect other than genuineness. To this end, we propose to generate a training set of GSCs from GGSs by transforming them using multiple autoencoders. The generated GSCs are used together with their original GGSs for training the discriminator. In our experiments, the proposed method achieved the recognition accuracy of up to 94% for several test datasets, which demonstrates the effectiveness and the generality of the proposed method.

1. Introduction

Biometrics is one of the hottest topics in the field of multimedia processing. Many verification and identification methods have been proposed in the past two decades, whose typical examples include face recognition [1], speaker recognition [2], fingerprint recognition [3], and so on. On the other hand, most of these biometric recognition methods generally suffer from spoofing attacks. Here, spoofing is an attacker’s malicious process to fool a recognition system using fake data by which he is recognized as a valid user. Hence, anti-spoofing methods have also been widely studied [4]–[6]. In recent years, the performance of multimedia generation techniques has been drastically improved with deep neural networks (DNNs), especially autoencoders (AEs) [7] and generative adversarial networks (GANs) [8], which allows us to automatically generate a fake of a target person’s face images, motion videos, and voice in a high level of quality. Babaguchi defined these fake multimedia data generated by computers as media clones and pointed out that there is a risk of spoofing attacks exploiting media clones [9]. Hence, it is desired to develop anti-spoofing methods that can discriminate media clones from genuine data. Several previous studies have already tackled this task, particularly focusing on voice [10], [11] and face images [12], [13].

Among a lot of biometric recognition methods, gait recognition is a relatively novel approach and attracts many researchers’ attention today. Especially, silhouette-based gait recognition has been actively studied due to its simplicity and convenience [14], [15]. Currently, there are only a few approaches of spoofing attacks against gait recognition because practical gait-based verification/identification systems have not been established yet. However, it will become a real threat in the near future because of the rapid growth and spread of multimedia generation techniques.

Gait recognition has been already applied to criminal investigation in some country [16], [17], which can be disturbed by multimedia generation techniques. For instance, if a criminal generates a sequence of fake images representing his walking motion by AEs or GANs and embeds it into publicly available videos (e.g. videos on YouTube), his location can be made uncleared. The similar situation can be occurred if a criminal transforms his appearance in a video into that of another person. In addition, when gait recognition will be applied to entrance management systems in the future, a computer-generated human motion video resembling a target person’s gait would cause a serious risk of spoofing attacks against the systems. In a general process of silhouette-based gait recognition, a system first extracts a sequence of human body silhouettes from a raw video and then identifies the extracted silhouette sequence. Between these two steps, if we can check whether the extracted silhouette sequence is given from an actually captured video or automatically generated one, we can reduce the above risk.

From the above backgrounds, in this paper, we focus on an anti-spoofing method for silhouette-based gait recognition systems. More specifically, we propose a supervised method for discriminating computer-generated gait silhouette videos from genuine ones observed from actual walking people. We refer to the generated silhouettes as gait silhouette clones (GSCs), which are an instance of media clones. Here, we consider that not only directly generated silhouettes but also those extracted from a synthetic video (e.g. a publicly available video containing automatically generated...
human regions) are kinds of GSCs.

In the past dozen years, a few studies have tackled the problem of spoofing attacks against gait recognition [18], [19]. However, their focus was not media clone-based spoofing but physical spoofing; that is, they only focused on the gait spoofing attack where an attacker physically tries to imitate clothes or walking style of a target person. Unlike them, we focus on the spoofing attack exploiting GSCs that are generated by multimedia generation techniques. In this sense, we believe that the existing methods and our proposed one can complement each other to make gait recognition systems more secure.

The main contribution of this paper is summarized as follows: First, this is the first work that indicates the risk of media clone-based spoofing attacks against gait recognition systems, to the best of our knowledge. Second, our proposed method has good generality; in other words, it can work well even when the video capturing environment for genuine gait silhouettes (GGSs) and generation methods for GSCs are different from those used in the training phase.

The remainder of this paper is organized as follows: After reviewing related work in Sect. 2, we propose a supervised method for discriminating GSCs from GGSs in Sect. 3. The performance of the proposed method is experimentally evaluated in Sect. 4, and finally, this paper is concluded in Sect. 5.

2. Related Work

A spoofing attack using information not directly captured from a living human is a serious problem for biometric recognition systems. To defend against the attack, it is important to examine whether the input to the systems has a liveness property or not. This task is called liveness detection. Methods of liveness detection for faces, fingerprints, and voice have been widely studied, which are deeply related to our target task. Hence, we first review them in Sect. 2.1. Next, in Sect. 2.2, we briefly review the existing anti-spoofing methods for gait recognition systems and clarify the difference of their target and ours. After that, we review the current status of generation techniques for human motion video in Sect. 2.3, which could be exploited for spoofing attacks against gait recognition systems in the near future.

2.1 Studies on Liveness Detection

The study of liveness detection for face recognition systems had been already started before DNN-based multimedia generation techniques were developed. Early methods mainly assumed a presentation attack in which attackers try to fool a camera-based face recognition system by presenting a valid user’s printed face image or replayed face video shown in a displaying device to the camera.

To deal with the presentation attack, several methods focused on optical flow field of face regions [20], [21]. In general, we can observe almost constant optical flow field from a video of two-dimensional objects including printed face images due to their planar shape. In contrast, we can observe more varying optical flow field from a video of living human’s face due to its three-dimensional shape. These properties are useful for examining whether the presented face is directly captured from a living human or not. Another strategy is to instruct a user in front of the system’s camera to perform some motion such as eye-blinking [22] and head motion [23]. Because an attacker presenting a printed face image or a replayed face video cannot follow the instruction, he is rejected by the system. These methods need a user’s cooperation, which could make the user uncomfortable. To avoid this problem, Zhang et al. focused on the reflectance property of human faces [24]. Human faces have a quite different reflectance property from that of other objects (silica gel, rubber, photo paper, and video screen) under invisible lights. Based on this fact, Zhang’s method shed invisible light to a user’s face and analyzed its reflectance to detect liveness property.

In recent years, researchers have assumed a spoofing attack by multimedia generation techniques as well as the presentation attack. To discriminate between genuine human faces and computer-generated fake faces, Conotter et al. focused on a physiological signal resulting from the human pulse [12]. Because of the human pulse, the surface color of living human faces periodically changes at high speed, which cannot be observed from computer-generated faces. Detecting the physiological signal, Conotter’s method successfully discriminated genuine faces from fake ones. As another strategy, Nguyen et al. proposed to use the difference between texture complexity of genuine faces and that of fake faces [13]. They found that images of living human faces have a complex texture caused by downy hairs, moles, wrinkles, and so on, which are difficult to be completely replicated. This difference is useful to discriminate between genuine and fake faces.

For fingerprint recognition, Marasco et al. gave a good survey on anti-spoofing schemes including liveness detection methods [6]. According to their survey, perspiration is an important evidence of living fingers. Due to the perspiration, gray level of living finger images changes over time. In contrast, images of fake fingers made by silicone resin or other similar materials have high uniformity over time. Derakhshani et al. utilized this property for detecting liveness of fingerprints [25]. Another good feature for fingerprint liveness detection is texture complexity. Marasco et al. said in their survey that materials used to make fake fingers such as silicon resin or gelatin consist of organic molecules tending to agglomerate. Therefore, the surface of a living finger is generally smoother than a fake finger. This causes a difference in texture complexity between images of living fingers and those of fake fingers. Moon et al. focused on this property and proposed a texture-based method of fingerprint liveness detection [26].

For speaker recognition, Shiota et al. proposed to use pop noise caused by human breath for voice liveness detection [10], [11]. When a living human speaks into a micro-
phone, his/her breath reaches it and is poorly reproduced by loudspeakers. This causes pop noise, which is difficult to be replicated. Hence, pop noise can be used as a good evidence of liveness property in voice data.

As reviewed above, anti-spoofing methods have been widely studied for face, speaker, and fingerprint recognition. However, for gait recognition, there are only a few anti-spoofing methods, which are reviewed in the next section.

2.2 Studies on Anti-Spoofing for Gait Recognition

The risk of spoofing attacks against gait recognition systems was firstly investigated by Gafurov et al. in 2007 [18]. Their focus was not silhouette-based gait recognition but accelerometer-based one, where an accelerometer sensor is attached to the hip of a user and its output, i.e., a time series of three-dimensional acceleration vectors, is used for recognition. Against this approach, they tested the risk of the physical spoofing attack where an attacker tries to mimic a target person’s walking style. Through their experimental results, they mainly concluded that it is difficult to spoof someone’s walking style to fool the accelerometer-based gait recognition systems.

More recently, Hadid et al. tested the risk of spoofing attacks against silhouette-based gait recognition systems [27]. Their assumed attack was also a physical one, where an attacker imitates a target person’s clothes to fool the system. In addition, they also assumed another kind of physical attacks where an attacker deliberately selects a valid user as a target whose body shape is similar to that of the attacker himself to increase the success possibility of his attack. They concluded that it is possible to spoof someone’s silhouettes, unlike the conclusion of Gafurov’s work. Thus, Hadid et al. proposed an anti-spoofing method in their another work [19]. This method first divides a given silhouette into multiple horizontal portions and next computes Local Binary Patterns from Three Orthogonal Planes (LBP-TOP) for each portion, which are then concatenated into a single histogram vector. Finally, histogram distance is calculated between two histogram vectors to recognize whether they are actually obtained from the same person or not. This is an extension of dynamic texture based gait recognition [28].

Unlike the above methods, our focus is the gait spoofing exploiting media clones, namely, automatically generated video of a walking person and its binarized (or silhouetted) version. These videos are expected to have a different characteristic from the video of physically mimicked gait. In this sense, the target of this work is different from that of the existing work, which can complement each other.

2.3 Studies on Motion Video Generation

We can easily obtain gait silhouettes from a video of walking person by binarizing it. Therefore, methods for automatically generating such a video will be in danger of being exploited for attacks against gait recognition systems. Existing methods for motion video generation are roughly divided into the following two categories: model-based and image-based. However, the former will be hardly exploited to the attacks because it needs detailed 3D human model of a valid user which is difficult to be obtained for attackers. Hence, we only focus on the latter, namely image-based methods.

Image-based methods do not assume any 3D model and directly generate a human motion video. Kobayashi et al. proposed to generate a human motion video from several short clips by seamlessly connecting 2D human regions in them [29]. Each of the short clips represents a motion primitive and the motion primitives can be periodically connected. Therefore, this method is suitable to generating periodic motions such as walking. Tüyük et al. focused on the 2D contour of a human region in a still image and proposed to transform the contour so that it traces some reference motion [30]. After the transformation, the texture in the original contour is mapped into the transformed one. More recently, there have been proposed several methods using GANs to generate a human image [31]–[33]. Given an original human image and a 2D skeleton data representing a certain target pose, these methods generate a new image in which the same person with the original one is taking the target pose. Although these methods only generate a single image at once, it is easy to generate multiple images from a sequence of poses and integrate them into a video. Moreover, Tulyakov et al. proposed a GAN-based video generation method called MoCoGAN [34], in which a human’s appearance and motion can be separately set by two different random vectors. This method allows us to generate a gait video of various people by using different vectors for the appearance and the fixed vector for the motion.

Currently, most of the above methods have a drawback: a lot of actual motion videos are required as a training data. However, their performance and practicality are still rapidly growing. We think misuse of these methods against gait recognition systems will become a real threat in the near future.

3. Supervised Method for Discriminating between Genuine and Cloned Gait Silhouettes

In this section, we make a detailed explanation of the proposed method that discriminates GSCs from GGSSs.

Before starting the explanation, we first make the definition of the genuineness of gait silhouettes. In this paper, only the silhouettes extracted from camera-captured video by hands or human region extraction techniques are defined as genuine. Hence, the silhouettes obtained by binarizing synthetic motion videos and those directly obtained as the output of deep networks are regarded as cloned.

As mentioned in Sect. 2.1, texture complexity plays an important role for discriminating computer-generated media clones from genuine data in the case of faces and fingerprints [13], [26]. If gait silhouettes have the similar property, in other words, if contour complexity of GGSSs differs from that of GSCs, we could develop an unsupervised method.
for our target task. However, contour complexity of GGSs depends on how accurately they were extracted from raw video or images. In addition, contour complexity of GSCs depends on their generation methods. Hence, we aim to develop a supervised method.

3.1 Overview

Let $Q$ denote a training dataset for training a recognizer that can discriminate GSCs from GGSs. $Q$ consists of $Q^f$ and $Q^c$, that is, $Q = Q^f \cup Q^c$, where $Q^f = \{V_i^f|i = 1, 2, \cdots\}$ is a set of videos of GGS and $Q^c = \{V_j^c|i = 1, 2, \cdots\}$ is a set of videos of GSC. $V_i^f$ and $V_j^c$ denote each gait silhouette video used for training the recognizer. $Q^f$ is collected by applying a human region extraction technique to raw videos and binarizing the extraction results, while $Q^c$ can be collected by multimedia generation techniques like the ones mentioned in Sect. 2.3. Once $Q^f$ and $Q^c$ are collected, current DNN technologies allow us to make a neural network $\mathcal{R}$ that can automatically extract a good feature from each video in $Q^f$ and $Q^c$ and successfully discriminate between them. Hence, how to collect $Q^f$ and $Q^c$ plays a key role in this strategy.

It is noted that $Q^f$ and $Q^c$ should not differ from each other in any aspect other than genuineness. For instance, suppose that $Q^f$ consists of three GGS videos $F$, $G$, and $H$ collected from actual walking people and $Q^c$ consists of three GSC videos $S$, $T$, and $U$ collected by a DNN-based generator, where the generators is trained with three GGS videos $S$, $T$, and $U$, as seen in Fig. 1. If we train a recognizer for discriminating between these $Q^f$ and $Q^c$, the trained recognizer does not work well; when a GGS video $U$ is input to the recognizer, it would judge “this video is cloned.” This is because $U$ is more similar to $U'$ than $F$, $G$, and $H$. In general, different people have different shapes of gait silhouettes, and similar people can generate a person’s GSCs that are very similar to his/her GGSs. Therefore, $S$, $T$, and $U$ are only slightly different from $S'$, $T'$, and $U'$, whereas they are significantly different from $F$, $G$, and $H$.

Based on the above consideration, we propose the following algorithm to collect $Q^f$ and $Q^c$ that are suitable to our target task:

1. Collect $Q^f$ by observing many walking people.
2. For each $V_i^f \in Q^f$, generate its cloned version as $V_i^c$ so that $V_i^f$ and $V_i^c$ do not differ from each other in any aspect other than genuineness.
3. Using $V_i^f$ generated in the step (2), construct $Q^c$ as $Q^c = \{V_i^c|i = 1, 2, \cdots\}$.
4. Train a recognizer network $\mathcal{R}$ using $Q = Q^f \cup Q^c$.

We first describe the required property for $V_i^f$ and $V_i^c$ in more detail in Sect. 3.2. Then we specify how to generate $V_i^f$ in Sect. 3.3.

3.2 Required Property for Training Dataset

Let $s_i$ be the $k$-th frame in $V_i$. Similarly, let $s_i$ be the $k$-th frame in $V_i^c$. Each $s_i$ and $s_i$ can be mainly determined by the following four factors: person, clothes, pose, and viewpoint. This is because we can always get the almost same silhouette image if we observe the same person who is wearing the same clothes and taking the same pose from the same viewpoint.

Let $f_{i,k}^o$, $f_{i,k}^p$, $f_{i,k}^{po}$, and $f_{i,k}^{vp}$ be feature vectors corresponding to person, clothes, pose, and viewpoint aspects of $s_i$, respectively. In addition, let $f_{i,k}$ be the concatenation of these four vectors, i.e.,

$$f_{i,k} = \left( f_{i,k}^o \right)^\top, \left( f_{i,k}^p \right)^\top, \left( f_{i,k}^{po} \right)^\top, \left( f_{i,k}^{vp} \right)^\top.$$

Using $f_{i,k}$ and an ideal image decoder $D_{\text{ideal}}$, which is mathematically a map from a feature vector to a silhouette image, we can formulate $s_{i,k}$ as

$$s_{i,k} = D_{\text{ideal}}[f_{i,k}] + \epsilon_{i,k},$$

where $\epsilon_{i,k}$ is an error term caused by noise in raw images, extraction error of human regions in raw video, and so on. Note that the effect of the error term on gait silhouette images is much smaller than that of the above factors.

Similar with $s_i$, we can formulate $s_{i,k}$ as

$$s_{i,k} = D_{\text{ideal}}[h_{i,k}] + \epsilon_{i,k},$$

where $h_{i,k}$ is the same type of vector as $f_{i,k}$ that represents person, clothes, pose, and viewpoint aspects of $s_{i,k}$. In fact, an actual image decoder $D$ is generally used for generating $s_{i,k}$, thus we can further formulate it as

$$s_{i,k} = D[h_{i,k}] + \epsilon_{i,k} = D[h_{i,k}].$$

Note that $\epsilon_{i,k}$ in Formulas (3) and (4) is an error term which is caused by the decoder $D$.

Because $\epsilon_{i,k}$ and $\epsilon_{i,k}$ are caused by different ways, their distributions differ from each other. The recognizer network $\mathcal{R}$ should capture this difference. To this end,

$$f_{i,k} = h_{i,k}$$

should be satisfied for all $i$ and $k$. This is the required property for a training dataset.
Hereafter, we omit \( i \) and \( k \) from the above notation when they do not have to be specified. (For instance, we use \( \epsilon^g \) instead of \( \epsilon_{i,k}^g \) in some cases.)

### 3.3 Training Data Generation Using Autoencoders

The required property described in the previous section can be satisfied by the following way: First, extract \( f_{i,k} \) from \( s_{i,k}^g \) using a certain encoder \( E \), namely,

\[
f_{i,k} = E[s_{i,k}^g].
\]  

(6)

Note that the encoder \( E \) is mathematically a map from a silhouette image to a feature vector. After that, generate \( s_{i,k}^c \) from the extracted feature by \( D \), namely,

\[
s_{i,k}^c = D[f_{i,k}] = D[E[s_{i,k}^g]].
\]  

(7)

Now our focus is on how to realize such an encoder \( E \) and a decoder \( D \). The feature vector \( f_{i,k} \) can be variously defined. However, regardless of the definition, it is quite labor-intensive to obtain the ground truth of \( f_{i,k} \) for all \( i \) and \( k \). To avoid this problem, we do not explicitly define \( f_{i,k} \) in the proposed method.

As mentioned in Sect. 3.2, the error term \( \epsilon^g \) gives much smaller effect on gait silhouette images compared with the four main factors, i.e., person, clothes, pose, and viewpoint. This means the variance of gait silhouette images caused by change in \( \epsilon^g \) is much smaller than that caused by change in \( f \). Hence, if we extract (nonlinear) principal components from a set of gait silhouette images, the extracted components are expected to be not correlated with the error term \( \epsilon^g \). Of course it is unclear which components are actually correlated with each of the four factors. There might be some components correlated with two or more factors. However, these are not matters as long as there are no components that are independent from all of the four factors.

Based on the above discussion, the proposed method trains autoencoders (AEs) using \( Q' \), whose encoder part and decoder part are used as \( E \) and \( D \), respectively. More specifically, we first divide \( Q' \) into several subsets and separately use each subset as training data to train multiple AEs, as seen in Fig. 2. Note that these subsets can be overlapped with each other. Then we input each \( s_{i,k}^g \) to the trained AEs and calculate \( s_{i,k}^c = D[E[s_{i,k}^g]] \) for all \( i \) and \( k \), resulting in \( Q' = \{V_i^c | i = 1, 2, \ldots \} \). For the loss function \( L \) of the AEs, we adopt mean-squared error, that is,

\[
L[E, D] = \sum_i \sum_k \left\| E[s_{i,k}^g] - D[E[s_{i,k}^g]] \right\|^2.
\]  

(8)

The number of the units in the intermediate layer, i.e., the dimension of the vector \( E[s^g] \), is set much smaller than the number of pixels in \( s \). These settings allow us to extract a kind of nonlinear principal components from \( s^g \) by \( E \), namely \( E[s^g] \), which is expected to be unrelated to the error term \( \epsilon^g \). \( Q' \) generated by the above procedure satisfies the requirement described in the previous section. In the remainder, we refer to each dimension of the extracted vector as a "Genuine" or "Cloned".
$E[s^g]$ as a latent feature. For the way to divide $Q^g$ into subsets, we employ bootstrap sampling. More specifically, we randomly select a number of gait silhouette images (or frames of video) from $Q^g$ to obtain a single subset, which is repeated enough times. Thanks to the bootstrap sampling we can generate a more diverse set of GSCs for training the recognizer $R$, which is expected to improve the generalization ability of the proposed method.

3.4 Training Procedure

Using $Q^g$ and $Q^c$ generated by the method of the previous section, we finally train the recognizer $R$.

Although we mentioned that $R$ should capture the difference between $e^g$ and $e^c$ in Sect. 3.2, they are only slightly different from each other. Rather than $e^g$ and $e^c$ themselves, their time-change is more informative. Hence, the proposed method first divides a given video into segments of $N$-frame length and then constructs $R$ that recognizes the genuineness of each segment, instead of processing the given video frame-by-frame. Finally, segment-wise recognition results are integrated into a unique label (genuine or cloned) by a majority voting rule. More specifically, the proposed method judges the given video as cloned when and only when $T_{\text{cloned}} > T_{\text{vote}}$, where $T_{\text{vote}} = \frac{T}{N}$ is the number of votes (or segments), $T$ is the total number of frames in the given video, and $T_{\text{cloned}}$ is the number of the segments that are recognized as cloned by $R$.

In the proposed method, we regard each of the above segments as an $N$-channel image, which is input to $R$. $R$ is designed as an ordinary convolutional neural network (CNN), whose structure is shown in Fig. 3. Softmax cross-entropy is adopted as the loss function of $R$.

4. Experiments

We conducted experiments to evaluate the performance of the proposed method. We first describe the experimental condition, especially the details of the employed datasets, in Sect. 4.1 and then discuss the results in Sects. 4.2. In Sects. 4.3 and 4.4, we conduct some additional analyses and report their results.

4.1 Experimental Setup

As a set of GGSs for use in the experiments, we employed OU-ISIR Gait Database [35], which consists of three sub-datasets: treadmill dataset A, B, and C. The treadmill dataset A includes gait silhouette videos of 34 people with speed variation from 2 [km/h] to 10 [km/h] at 1 [km/h] interval. Each person walks twice for each speed, thus the total number of the videos is $34 \times 9 \times 2 = 612$. This dataset has less variety of clothes. On the other hand, the tread-

![Network structure of three AEs used for generating GSCs from GGSs in the experiments.](image-url)

The meanings of Conv., MP, FC, and KS are same with those in Fig. 3. Deconv. and UNP means a deconvolution layer and an unpooling layer, respectively.
mill dataset B includes gait silhouette videos of 68 people who are wearing 32 kinds of clothes. The total number of the videos is $68 \times 32 = 2176$. This dataset has less variety of walking speed contrary to the dataset A. The treadmill dataset C consists of gait silhouette videos of 185 people with various gait fluctuations among periods, which means people change their walking speed within a single video clip. Since each person walks twice in this dataset, the total number of videos is $185 \times 2 = 370$. We regarded all of these 3158 ($= 612 + 2176 + 370$) videos as GGSs and randomly divided them into three disjoint parts, which we call Train-GGS, Train-GGS-Another, and Test-GGS-Closed. Train-GGS and Train-GGS-Another contain a quarter of the 3158 GGSs, respectively, and Test-GGS-Closed contains the remaining half.

Next, we converted all videos in Train-GGS, Train-GGS-Another, and Test-GGS-Closed into their cloned versions using AEs in order to obtain a set of GSCs. The structures of the AEs designed for this purpose are shown in Fig. 4. Each of these three AEs was separately trained three times using a subset of Train-GGS, which consists of 90,000 silhouette images randomly selected by bootstrap sampling for each trial. This means we prepared nine different AEs in total in this experiment. Figure 5 shows an example of GGS and its cloned versions generated by nine AEs.

![Fig. 5](image)

constructed another set of GGSs and GSCs which is used as a test dataset. For GGSs, we captured videos of 14 walking people from the side view and extracted their silhouettes via the Chroma-Key process. Since we did not use a treadmill, the position of the extracted silhouettes moves over time in the captured videos. Therefore, we normalized the position of the silhouettes by translating them so that the center of the head region is fixed. In this sense, our GGS capturing environment largely differs from that of the OU-ISIR Gait Database. We repeated the above capturing process 13 times for each person, thus the total number of the captured videos is $13 \times 14 = 182$. We refer to this video set as Test-GGS-Open.

For GSCs, we employed a GSC-generator proposed in our previous work [36] that is totally different from the above nine AEs; it can clone a gait silhouette video of a target person from his/her single silhouette image. In [36], we designed this generator with the following idea: a gait silhouette image can be reconstructed by its phase component and shape component. The former is a real value ranging from 0 to $2\pi$ representing the pose aspect of the silhouette. This setting is based on the fact that walking is a periodic motion. On the other hand, the latter, i.e., shape component, is a real vector representing person, clothes, and viewpoint aspects of the silhouette, which is common for all frames in a single video. Based on the idea, the GSC-generator first extracts a shape component from the input image and uses it together with an arbitrary sequence of phase components to generate an output video. With this generator, we actually generated 306 videos from the randomly selected frames in the treadmill dataset A. We refer to this video set as Test-GSC-Open.

4.2 Results and Discussion

Figure 6 shows the recognition accuracy of $R$ and that of $R'$ on Test-GGS-Closed and Test-GSC-Closed with several settings of $N$. Note that $N$ is the length of the segments input to $R$ and $R'$, defined in Sect. 3.4. When $N \geq 2$, $R$ can discriminate between GGSs and GSCs with the accuracy of more than 80% on an average. The highest accuracy of 94.9% is obtained with $N = 4$. In contrast, the accuracy of $R'$ is less than 60% in most cases. This result indicates the effectiveness of the proposed method.

It is pointed out that the accuracy of $R$ tends to be lower with larger $N$. This is because the total number of votes becomes smaller with larger $N$ in the process of the majority voting described in Sect. 3.4. For instance, from a gait silhouette video whose length is 100 frames, we can obtain 50 segments in total when $N = 2$ by dividing the video into 2-frame segments, while we can obtain only 10 segments in total when $N = 10$. Hence, the reliability of the majority voting becomes lower with larger $N$, which degrades the final recognition accuracy. This can be confirmed by the fact that the accuracy of $R$ before the majority voting is not so degraded with larger $N$ in Fig. 7.

Interestingly, with $N = 1$, not only $R'$ but also $R$ does
Fig. 6 Recognition accuracy on Test-GGS-Closed and Test-GSC-Closed. Red lines indicate the results of the proposed recognizer, \( R \).

Fig. 7 Recognition accuracy before majority voting for Test-GGS-Closed and Test-GSC-Closed. 

not work well. This means it is difficult to discriminate between GGSs and GSCs from a single frame. As mentioned in Sect. 3.2, the distribution of \( \epsilon^g \) differs from that of \( \epsilon^c \). However, their difference is very slight and difficult to be captured even by a DNN-based recognizer. In contrast, time-change of \( \epsilon^g \) is sufficiently different from that of \( \epsilon^c \), which allows \( R \) to successfully discriminate GSCs from GGSs when \( N \geq 2 \). We consider the reason as follows. A main factor causing \( \epsilon^g \) in a GGS is image noise, which is differently produced frame-by-frame. Hence, \( \epsilon^g \) does not smoothly change over time. On the other hand, \( \epsilon^c \) in a GSC lacks such a frame level randomness because it is totally caused by computer processing. Therefore, when a decoder \( D \) generates two consecutive frames whose silhouette shapes are similar, the error terms corresponding to these two frames would also be similar. As a result, \( \epsilon^c \) smoothly changes over time contrary to \( \epsilon^g \). This difference can be successfully captured by \( R \).

Next, we show the recognition accuracy of \( R \) and that of \( R' \) on Test-GGS-Open and Test-GSC-Open in Fig. 8. In this figure, we can see the similar tendency with Fig. 6; the accuracy of \( R' \) is around 50% in most cases whereas the accuracy of \( R \) is more than 80% on an average with \( N \geq 2 \), although the way to collect Test-GGS-Open and Test-GSC-Open is totally different from that to collect the training data, i.e., Train-GGS and Train-GSC. \( R \) achieves the highest accuracy of 94.3% with \( N = 2 \), which is comparable to the highest accuracy for Test-GGS-Closed and Test-GSC-Closed shown in Fig. 6. This result indicates that the proposed method has good generality.

As mentioned in Sect. 3.3, the proposed method uses multiple AEs trained with various subsets of \( Q^g \) to construct \( Q^c \). To experimentally test the effectiveness of this process, we evaluated the recognition accuracy of \( R \) under the following four conditions, and then compared the results: (i) training a single AE with a single subset, (ii) training a single AE three times with three different subsets, (iii) training three AEs with a single subset, and (iv) training three AEs three times with three different subsets. The results are shown in Fig. 9 and 10, where a better performance is achieved with conditions (ii) and (iv) than conditions (i) and (iii). The highest recognition accuracy is obtained with condition (iv). This result demonstrates the effectiveness of using various subsets.

Now we can draw the following conclusion from all the results shown above. First, for discriminating between GSCs and GGSs, it is important to generate \( Q^c \) that does not differ from \( Q^g \) in any aspect other than genuineness. Second, it can much increase the recognizer’s generality to use multiple AEs trained with various subsets of \( Q^g \) for constructing \( Q^c \).

4.3 Hypothesis Verification

In Sect. 3, we introduced two hypotheses. One is that the distribution of \( \epsilon^g \) differs from that of \( \epsilon^c \). The other is that the latent features extracted by AEs are correlated with at least one of the four main factors, namely, person, clothes, pose, and viewpoint. To quantitatively test these hypothesis, we conducted additional analyses, whose results are reported in this section.
4.3.1 Analysis on the Error Term

As seen in Formulas (2) and (3), the error term is defined as the difference between an silhouette image and its noise-free version obtained with the ideal decoder $D_{\text{ideal}}$. However, since there is no $D_{\text{ideal}}$ in real environments, we cannot experimentally compute $\epsilon^d$ and $\epsilon^s$. This means it is impossible to directly analyze the error term. Therefore, we analyzed the difference between GGSs and their cloned version generated by AEs, instead of a direct analysis. For this purpose, we input every silhouette image $s^d$ in the treadmill dataset $B$ into the AEs shown in Fig. 4 (a) and (b). Let $s^a$ and $s^b$ be the GSCs generated with these two AEs from $s^d$. Since the four main factors in $s^d$ are expected to be kept in $s^a$,

$$s^d - s^a = (D_{\text{ideal}}[f] + \epsilon^d) - (D_{\text{ideal}}[f] + \epsilon^s) = \epsilon^d - \epsilon^s, \quad (9)$$

is satisfied. This is also the case with $\epsilon^s$.

First, we analyzed the mean of $\epsilon^d - \epsilon^s$, i.e., $\mathbb{E}[\epsilon^d - \epsilon^s]$. If this is not zero, we can conclude that there is a difference between the distribution of $\epsilon^d$ and that of $\epsilon^s$. The result is shown on the left side of Fig. 11. Black, gray, and white regions in this figure indicate $\mathbb{E}[s^d(x) - s^a(x)] < 0$, $\mathbb{E}[s^d(x) - s^a(x)] = 0$, and $\mathbb{E}[s^d(x) - s^a(x)] > 0$, respectively, where $s^d(x)$ and $s^a(x)$ are the pixel value of pixel $x$ in $s^d$ and that in $s^a$. Since we multiplied each pixel value by eight for visibility, the actual difference between $s^d$ and $s^a$ is much smaller. However, this result clearly demonstrates that $\mathbb{E}[\epsilon^d - \epsilon^s]$ is not zero.

Second, we analyzed the distribution of $||s^d - s^a|| - ||s^b - s^a||$. If $||s^b - s^a||$ tends to be smaller than $||s^d - s^a||$, we can conclude that the distribution of GSC’s error term has a common characteristic regardless of the kind of the decoder $D$, which is not common in the distribution of GGS’s error term. The result is shown on the right side of Fig. 11, where the value of $||s^d - s^a|| - ||s^b - s^a||$ tends to be larger than 0. This clearly demonstrates that $||s^b - s^a||$ tends to be smaller than $||s^d - s^a||$.

The results of the above two analyses would be an evidence of our hypothesis on the error term.

4.3.2 Analysis on the Latent Features Extracted with AE

To analyze the latent features extracted with AE, we used the treadmill dataset $B$ again. Since this dataset contains gait silhouette videos of 68 different people wearing 32 kinds of clothes, and each video consists of a sequence of images with various poses, it is suitable to the analysis. Note that we ignored the viewpoint factor and only focused on person, clothes, and pose factors in this analysis, because all videos in the treadmill dataset $B$ are captured from the same viewpoint.

As previously mentioned, the treadmill dataset $B$ contains $68 \times 32 = 2176$ videos, where 68 and 32 are the number of person-classes and that of clothes-classes. This means there are 2176 person-clothes-classes, which are the direct product set of the person-classes and the clothes-classes. Based on this property, we extracted 32-dimensional latent features from every frame of the 2176 videos with the AE shown in Fig. 4 (a), and then calculated the following four kinds of variances for each feature: the total variance $\nu_1(d)$, the within-class variance of the person-classes $\nu_p(d)$, the within-class variance of the clothes-classes $\nu_c(d)$, and the within-class variance of the person-clothes-classes $\nu_{pc}(d)$. Here, $d (1 \leq d \leq 32)$ means the ID of each latent feature.

Since $\nu_{pc}(d)$ does not include between-class variance of the person-classes and that of the clothe-classes, it becomes large if and only if the $d$-th feature is strongly co-
related with pose factor. In contrast, $v_c(d)$ additionally includes between-class variance of the person-classes, thus it becomes large when $d$-th feature is strongly correlated with pose and person factors. Similarly, $v_p(d)$ becomes large when $d$-th feature is strongly correlated with pose and clothes factors. Hence, comparing the value of these four variances allows us to examine whether the $d$-th feature is strongly correlated with each of the three factors or not.

The result of the analysis is shown in Fig. 12. We can see from the figure that there are no features whose total variance is significantly small ($\min_{d}(\sum_{i}v_{c}(d)) = 0.582$). This demonstrates our second hypothesis that all latent features extracted with AE are strongly correlated with at least one factor. Especially, pose factor is strongly correlated with all features ($\min_{d}(\sum_{i}v_{p}(d)) = 0.510$). This is because the shape of a gait silhouette most widely varies with change in pose. We can find that $v_p(d)$ is larger than $v_c(d)$ for most features. This is because different clothes sometimes yield quite large change in gait silhouette (e.g. long skirts vs. pants).

Based on the result shown in Fig. 12, we can classify the latent features into several groups as below.

- **Feature ID 3, 5, 9, 10, 11, 17, 20, 23, 28, 31, and 32.** For these features, $v_{pc}(d)$ only slightly smaller than $v_c(d)$, i.e., $v_{pc}(d) \approx v_c(d)$. This means they are only correlated with pose factor.
- **Feature ID 1, 2, 6, 12, 13, 14, 15, 16, 18, 19, 21, 25, and 26.** For these features, we can see $v_{pc}(d) < v_p(d) \approx v_t(d)$. This means they are strongly correlated with clothes factor as well as pose factor.
- **Feature ID 7, 22, 24, and 29.** For these features, we can see $v_{pc}(d) > v_p(d)$. This means they are more strongly correlated with person factor than clothes factor.
- **Feature ID 4, 8, 27, and 30.** For these features, we can see $v_{pc}(d) < v_c(d) \approx v_p(d) < v_t(d)$. This means they are strongly correlated with all the three factors.

### 4.4 Visual Evaluation of the Trained Recognizers

To find out which part in a gait silhouette more contributes to the discrimination between GGSs and GSCs, we visually evaluated $R$ and $R'$ using Grad-CAM [37] and tried to highlight the regions judged as contributing part by the two recognizers. Note that, as a target of Grad-CAM, we used the last convolutional layer of the network shown in Fig. 3. Figure 13 shows the results for several samples in Test-GGS-Open, in which red and green regions were judged as contributing while blue regions were not.

As seen in Fig. 13, whole region in the silhouette image is blue for both $R$ and $R'$ in the case of $N = 1$. This means the recognizers cannot discriminate between GGSs and GSCs at all. In contrast, in the case of $N \geq 2$, there are several red and green regions, whose characteristic is different for $R$ and $R'$. In the result of $R'$, it depends on individual samples which regions are red or green. This means $R'$ cannot extract a good feature that is commonly useful for checking any gait silhouette video. On the other hand, in the result of $R$, the hip region and its right neighboring region are always red or green, which allows $R$ to extract a commonly useful feature from these regions. This is why $R$ can achieve higher performance than $R'$ with $N \geq 2$, as seen in Fig. 6, 7, and 8.

In the side view gait silhouettes, the region neighboring the hip strongly reflects the shape of the upper wear (e.g. coattail), which can be changed by not only the four main factors described in Sect. 3.2 but also a weak force of wind pressure. Since this force complexly changes over time, $\epsilon$ around the hip region does not smoothly change in GGSs. This is opposite to the property of $\epsilon$ in GSCs mentioned in

![Fig. 12](image1.png) Four kinds of variances of latent features extracted with AE.

![Fig. 13](image2.png) Examples of visual evaluation results based on Grad-CAM.
Sect. 4.2. Hence, the hip region and its neighboring region are commonly useful for discriminating between GGSs and GSCs. To fool anti-spoofing methods like the proposed one, attackers have to clone the detailed shape of clothes (especially upper wears) and their time-change. However, they are not important for fooling gait recognition systems, because clothes-independent features are desirable and often employed in gait recognition. Hence, it is much more difficult for the attackers to fool both the gait recognition systems and the anti-spoofing methods simultaneously than to fool either of them. This indicates the beneficialness of the anti-spoofing methods like the proposed one.

5. Conclusion

This paper proposed a method for discriminating between GGSs and GSCs, considering the future risk of spoofing attacks against gait recognition systems. The proposed method evaluates the genuineness of a given video segment-by-segment, whose results are integrated by a majority voting method. The proposed method compares the genuine silhouette video with the spoofed one, and determines the genuineness of the given video segment. The method can complement general gait recognition systems that utilize cloth-independent features. The findings of this study are summarized as follows:

- To construct a training set of GSCs, it is desirable to train multiple AEs with various subsets of a given training set of GGSs and that of GSCs separately. This demonstrates the effectiveness and the generalizability of the proposed method.

The detailed shape of clothes and their time-change are a good feature that makes the performance higher. This can complement general gait recognition systems that utilize cloth-independent features.

In this paper, we only considered the gait silhouettes directly generated by deep networks in the experiments. However, in practice, the silhouettes obtained by binarizing a synthetic human motion video should also be considered. To verify the robustness of the proposed method to such silhouettes is an important future work. In addition, we will extend the proposed method so that it can handle the time-change of the silhouette shape in a more sophisticated way.

References


Yuki Hirose received the B.E. degree in Engineering Science from Osaka University in 2018. His research interests include gait anonymization and face image generation. He is a member of IEICE.

Kazuo Nakamura received the B.S. degree in Engineering from Kyoto University in 2005, and the M.S. and Ph.D. degrees in Informatics from Kyoto University in 2007 and 2011, respectively. He is currently an Assistant Professor at Graduate School of Engineering, Osaka University, from 2012. His research interests include image processing, image recognition, and video analysis. He is a member of IEEE, IEICE, IPSJ, and ITE.

Naoko Nitta received the B.E., M.E., and Ph.D. degrees in Engineering Science from Osaka University in 1998, 2000, and 2003, respectively. She is currently an Associate Professor in Graduate School of Engineering, Osaka University. From 2002 to 2004, she was a research fellow of the Japan Society for the Promotion of Science. From 2003 to 2004, she was a Visiting Scholar at Columbia University, New York. Her research interests are in the areas of video content analysis and image/audio processing. She received Best Paper Award of 2006 Pacific-Rim Conference on Multimedia (PCM2006). She is a member of IEEE, ACM, IEICE, and ITE.

Noboru Babaguchi received the B.E., M.E., and Ph.D. degrees in communication engineering from Osaka University, in 1979, 1981 and 1984, respectively. He is currently a Professor of the Department of Information and Communications Technology, Osaka University. From 1996 to 1997, he was a Visiting Scholar at the University of California, San Diego. His research interests include gait anonymization and face image generation. He has published over 250 journal and conference papers and several textbooks. Dr. Babaguchi received Best Paper Award of PCM2006 and IAS2009. He is on the editorial board of Multimedia Tools and Applications, and New Generation Computing. He served as a Guest Editor of IEEE TIFS, Special Issue on Intelligent Video Surveillance for Public Security & Personal Privacy. He also served as a workshop Co-chair of MIR2001, a Track Co-chair of IEEE ICME2006, a General Co-chair of MMM2008, a Sub-Track Chair of APSSIP ASC 2009, a General Co-Chair of ACM Multimedia 2012, a Track Co-Chair of ICPR2012, an Area Chair of IEEE ICME2013, and an Honorary Co-Chair of ACM ICMR2018. He also served on program committee of international conferences in multimedia related fields. He is a Fellow of IEICE, a Senior Member of IEEE, and a member of ACM, IPSJ, ITE and JSIA.