GenFace: Improving Cyber Security Using Realistic Synthetic Face Generation

Margarita Osadchy[1(✉)], Yan Wang[2], Orr Dunkelman[1], Stuart Gibson[2], Julio Hernandez-Castro[3], and Christopher Solomon[2]

[1] Computer Science Department, University of Haifa, Haifa, Israel
{rita,orrd}@cs.haifa.ac.il
[2] School of Physical Sciences, University of Kent, Canterbury, UK
{s.j.gibson,c.j.solomon}@kent.ac.uk
[3] School of Computing, University of Kent, Canterbury, UK
J.C.Hernandez-Castro@kent.ac.uk

Abstract. Recent advances in face recognition technology render face-based authentication very attractive due to the high accuracy and ease of use. However, the increased use of biometrics (such as faces) triggered a lot of research on the protection biometric data in the fields of computer security and cryptography.

Unfortunately, most of the face-based systems, and most notably the privacy-preserving mechanisms, are evaluated on small data sets or assume ideal distributions of the faces (that could differ significantly from the real data). At the same time, acquiring large biometric data sets for evaluation purposes is time consuming, expensive, and complicated due to legal/ethical considerations related to the privacy of the test subjects. In this work, we present GenFace, the first publicly available system for generating synthetic facial images. GenFace can generate sets of large number of facial images, solving the aforementioned problem. Such sets can be used for testing and evaluating face-based authentication systems. Such test sets can also be used in balancing the ROC curves of such systems with the error correction codes used in authentication systems employing *secure sketch* or *fuzzy extractors*. Another application is the use of these test sets in the evaluation of privacy-preserving biometric protocols such as GSHADE, which can now enjoy a large number of synthetic examples which follow a real-life distribution of biometrics. As a case study, we show how to use GenFace in evaluating SecureFace, a face-based authentication system that offers end-to-end authentication and privacy.

Keywords: Synthetic face generation · GenFace · SecureFace · Biometrics · Face-based authentication · Face verification

1 Introduction

Biometric systems have become prevalent in many computer security applications, most notably authentication systems which use different types of biometrics, such as fingerprints, iris codes, and facial images.

S. Dolev and S. Lodha (Eds.): CSCML 2017, LNCS 10332, pp. 19–33, 2017.
DOI: 10.1007/978-3-319-60080-2_2

A central problem in developing reliable and secure biometric systems is the fuzziness of biometric samples. This fuzziness is caused by the sampling process and by the natural changes in appearance. For example, two images of the same face are never identical and could change significantly due to illumination, pose, facial expression, etc. To reduce these undesirable variations in practical applications, images are usually converted to lower-dimensional representations by a feature extraction process. Unfortunately, no representation preserves the original variation in identity while cancelling the variation due to other factors. As a result, there is always a trade-off between the robustness of the representation (consistency of recognition) and its discriminating power (differentiating between different individuals).

Devising robust features requires a significant number of training samples—the most successful systems use several million images (see [19] for a more detailed discussion). Most of these samples are used for training, while evaluation (testing) of the system is typically done on a relatively small dataset. Since such systems may need to support large sets of many users (such as border control systems or national biometric databases including the entire population[1]), a small-scale evaluation is insufficient when testing the system's consistency in large-scale applications.

The solution seems to be the testing of the system on a large dataset. However, acquiring large data sets is difficult due to several reasons: The process of collecting the biometric samples is very time consuming (as it requires many participants, possibly at different times) and probably expensive (e.g., due to paying the participants or the need to annotate the samples). Moreover, as biometric data is extremely private and sensitive (e.g., biometric data cannot be easily changed), collecting (and storing) a large set of samples may face legal and ethical constraints and regulations.

A possible mitigation to these problems is working with synthetic biometrics—"biometrics" synthetically generated from some underlaying model. Given a reasonable model, which approximates the real-life biometric trait, offering efficient sampling procedures, with sufficiently many "possible samples", designers of biometric systems can query the model, rather than collecting real data. This saves both the time and the effort needed for constructing the real dataset. It has the additional advantage of freeing the designer from legal and ethical constraints. Moreover, if the model is indeed "rich" enough, one can obtain an enormous amount of synthetic biometric samples for testing and evaluate large-scale systems.

In the case of fingerprints, the SFinGe system [4] offers the ability to produce synthetic fingerprints. SFinGe can generate a database of such synthetic fingerprints, which can be used for the training and evaluation of fingerprints' biometric systems. Fingerprint recognition algorithms have been shown to perform equally well on the outputs of SFinGe and on real fingerprint databases.

In the case of faces, some preliminary results were discussed in [24]. The aim of [24] was to transform one sensitive data set (which contains real users) into a

[1] Such as Aadhaar, the Indian biometric database of the full Indian population.

fixed, secondary data set of synthetic faces of the same size. The transformation was based on locating close faces and "merging" them into new synthetic faces. This approach has strong limitations, both in the case of security (as one should perform a very thorough security analysis) as well as usability (as there is only a single, fixed size, possible data set). Finally, [24] does not offer a general purpose face generation system and it is not publicly available.

In this paper, we introduce GenFace, system that generates synthetic faces and is publicly available. GenFace allows the generation of many synthetic facial images: the user picks how many "identities" are needed, as well as how many images per "identity" to generate. Then, using the active appearance model [7], the system generates the required amount of synthetic samples. Another important feature of our system is the ability to control the "natural" fuzziness in the generated data.[2]

Synthetic faces created by GenFace can be used in evaluating and testing face-based biometric systems. For example, using GenFace it is possible to efficiently evaluate and determine the threshold that face recognition systems should set in distinguishing between same identity and different ones. By varying this threshold, one can obtain the ROC curve of the system's performance, which is of great importance in studying the security of an authentication system.

Similarly, a large volume of work in privacy-preserving techniques for biometrics (such as fuzzy commitment [14] and fuzzy extractors [9]) treat the fuzziness using error correction codes. Such solutions and most notably, systems using such solutions (e.g., [5,6,10,26]) and their efficiency rely on the parameter choices. Our system offers face generation with a controlled level of fuzziness, which allows for a more systematic evaluation of these protection mechanisms and systems.

We note that GenFace can also be used in other contexts within computer security. Protocols such as GSHADE [3], for privacy-preserving biometric identification, should be evaluated with data which is similar to real biometrics. By using the output of GenFace, such protocols could easily obtain a large amount of synthetic samples 'at the press of a button', which would allow for more realistic and accurate simulation of real data, without the need for collecting and annotating the dataset.

Finally, GenFace can be used in evaluating large scale face-based biometric systems. For example, consider representations (also called templates) which are too short to represent a large set of users without collisions, but are sufficiently long to represent a small set of users without such collisions. It is of course better to identify such problems *before* the collection of millions of biometric samples, which can be easily done using synthetic biometrics.

This paper is organized as follows: Sect. 2 introduces GenFace and how it produces synthetic faces. In Sect. 3 we show that a real system, *SecureFace* [10], reaches similar accuracy results using synthetic faces and using real faces and its evaluation on a large-scale data set, generated by GenFace. Section 4 concludes our contribution and discusses future research directions.

[2] Changes in viewing conditions require the use of a 3D model and will be considered in future work.

2 GenFace System for Synthetic Faces

The main function of the GenFace System is generating a large number of face images of the same or different identities. It generates a (user) specified number of random, different, facial identities (face images) which we refer to as *seed points*. A set of faces, comprising subtle variations of a seed face can then be generated. The difference (distance to the seed point) in appearance between the images comprising the set can be controlled by the user. Below a certain distance threshold the differences will be sufficiently small such that faces in the set, which we refer to later as *offspring* faces, have the same identity as the seed.

In this section, we first describe the generative model that we use for sampling synthetic faces. We then introduce our specific methods for sampling seed faces and for sampling offspring faces. We then present the GenFace user interface and some guidelines for using our system.

2.1 Model for Representing Facial Appearance

Different models have been proposed for generating and representing faces including, active appearance models [7], 3D deformable models [2], and convolutional neural networks [18,30]. Here we use an appearance model (AM) [7] due to its capacity for generating photo-realistic faces (e.g. [12,23]). The representation of faces within an AM is consistent with human visual perception and hence also compatible with the notion of face-space [28].[3] In particular, the perceptual similarity of faces is correlated with distance in the AM space [17].

AMs describe the variation contained within the training set of faces, used for its construction. Given that this set spans all variations associated with identity changes, the AM provides a good approximation to any desired face. The distribution of AM coefficients (that encode facial identity) of faces belonging to the same ethnicity are well approximated by an independent, multivariate, Gaussian probability density function [20,27,29].

We follow the procedure for AM construction, described in [12]. The training set of facial images, taken under the same viewing conditions, is annotated using a point model that delineates the face shape and the internal facial features. In this process, 22 landmarks are manually placed on each facial image. Based on these points, 190 points of the complete model are determined (see [12] for details). For each face, landmark coordinates are concatenated to form a shape vector, x. The data is then centered by subtracting the mean face shape, $\bar{x}$, from each observation. The shape principle components P_s are derived from the set of mean subtracted observations (arranged as columns) using principal components analysis (PCA). The synthesis of a face shape, denoted by $\hat{x}$, from the *shape model* is achieved as follows,

$$\hat{x} = P_s b_s + \bar{x}, \tag{1}$$

[3] Hereafter we use the term face-space to mean the space spanned by a set of principal components, derived from a set of training face images.

where b_s is a vector in which the first m elements are normally distributed parameters that determine the linear combination of shape principal components and the remaining elements are equal to zero. We refer to b_s as the *shape coefficients*.

Before deriving the texture component of the AM, training images must be put into correspondence using non-rigid shape alignment procedure. Each shape normalized and centered RGB image of a training face is then rearranged as a vector g. Such vectors for all training faces form a matrix which is used to compute the texture principle components, P_g, by applying PCA. A face texture, denoted by $\hat{g}$, is reconstructed from the *texture model* as follows,

$$\hat{g} = P_g b_g + \bar{g}, \tag{2}$$

where b_g are the *texture coefficients* which are also normally distributed and $\bar{g}$ is the mean texture.

The final model is obtained by a PCA on the concatenated shape and texture parameter vectors. Let Q denote the principal components of the concatenated space. The AM coefficients, c, are obtained from the corresponding shape, x, and texture, g, as follows,

$$c = Q^T \begin{bmatrix} rb_s \\ b_g \end{bmatrix} = Q^T \begin{bmatrix} wP_s^T(x - \bar{x}) \\ P_g^T(g - \bar{g}) \end{bmatrix} \tag{3}$$

where w is a scalar that determines the weight of shape relative to texture.

Generating new instances of facial appearance from the model requires a sampling method for AM coefficients, c, which is described in detail in Sect. 2.2. The shape and texture coefficients are then obtained from the AM coefficients as follows: $b_s = Q_s c$ and $b_g = Q_g c$, where $[Q_s^T Q_g^T]^T$ is the AM basis. The texture and shape of the face are obtained via Eq. (1) and (2) respectively. Finally, the texture $\hat{g}$ is warped onto the shape $\hat{x}$, resulting in a face image.

The identity change in the face-space is a slowly changing function. Thus, there is no clear border between different identities. However, given the success of numerous face recognition methods, which rely on the Euclidean distance in the face-space to make recognition decisions (e.g., [1,11,25]), we can conclude that Euclidean distance in the face-space is correlated with the dissimilarity of the corresponding facial images. Another conclusion that can be drawn from the success of the face-space recognition methods, is that face-space is isotropic.[4] Based on these two observations, we suggest simulating the variation in appearance of the same person by sampling points in a close proximity to each other. Specifically, we define an identity as a random seed in the face-space and generate different images of this identity by sampling points in the face-space at a distance s from the seed point. Here s is a parameter of the system which is evaluated in our experiments (in Sect. 3).

2.2 Sampling AM Coefficients for Seed Faces

Since AM coefficients follow a multivariate Gaussian distribution, most of the faces in AM representation are concentrated near the hyper-ellipsoid with radii

[4] Our experiments, reported in Sect. 3.2 verify these assumptions.

approximately equal to $\sqrt{d}$ standard deviation units, where d is the dimension of the face-space. This follows from the fact that in high dimensional space, the distance between the samples of a multivariate Gaussian distribution to its mean follows the chi-distribution with an average of about $\sqrt{d}$ and variance that saturates at 0.7 (for $d > 10$). Since the standard deviation of the corresponding chi-distribution is narrow, sampling on the surface of the hyper-ellipsoid retains most of the probabilistic volume (i.e., most of the "plausible" faces in this face-space[5]). Also, it prevents the caricature effect associated with the displacement of a face in the direction directly away from the origin [13,16].

Let $N(0, \sigma)$ denote the distribution of AM coefficients where $\sigma = [\sigma_1, \sigma_2, \cdots, \sigma_d]$ are the standard deviations of the face-space. Let V denote the hyper-ellipsoid in R^d with radii $\sqrt{d}\sigma_i$ and ∂V denote its surface. To ensure the diversity of synthesized faces with respect to identity change, we should sample seed faces on ∂V uniformly at random. To this end, we implemented an algorithm from [21] that offers a method for uniform sampling from a manifold enclosing a convex body (summarized in Algorithm 1). Given a confidence level $1 - \epsilon$, the algorithm first draws a point $\boldsymbol{x}$ from the volume V uniformly at random, and then simulates a local diffusion process by drawing a random point p from a small spherical Gaussian centered at $\boldsymbol{x}$. If p is located outside V, the intersection between ∂V and a line segment linking p with $\boldsymbol{x}$ will be a valid sample, the distribution of which has a variation distance $O(\epsilon)$ from the uniform distribution on ∂V. It is essential to draw uniform samples efficiently from within V, thus we have also implemented an efficient sampler for this purpose, as proposed in [8].

2.3 Sampling AM Coefficients for Offspring Faces

Once a seed face has been uniformly sampled from ∂V, a set of offspring faces can be sampled from a local patch of ∂V around the seed face. Given a sampling distance s, the system can randomly pick up an offspring face located on ∂V which is at a distance s away from the seed face. If s is sufficiently small, all offspring faces will correspond to the same identity as the seed face.

Before describing the algorithm for sampling offspring face coefficients, we introduce the following notations: Let $P : \{p | p \neq \mathbf{0}\} \rightarrow \{\hat{p} | p \in \partial V\}$ be a mapping function that projects any point in the face-space, except the origin, onto ∂V by rescaling: $\hat{\boldsymbol{p}} = \frac{k}{\|M^{-1}\boldsymbol{p}\|}\boldsymbol{p}$, where M is a $d \times d$ diagonal matrix of standard deviations σ_i $(i = 1, .., d)$ and $k \approx \sqrt{d}$.

To provide diversity among offspring faces, we sample them along random directions. Given a seed face $\boldsymbol{x}$ and a distance s, the algorithm (summarized in Algorithm 2) repeatedly samples a face $\hat{\boldsymbol{x}}$ on ∂V at random until its distance to $\boldsymbol{x}$ exceeds s. The vector $\hat{\boldsymbol{x}} - \boldsymbol{x}$ defines a sampling direction for an offspring face. The algorithm proceeds by finding a point in this direction such that its

[5] It is unclear what should be the training size of a face-space that models all possible faces. However, we note that a face which is not "plausible" in some face-space, i.e., is very far from the surface of the face-space is likely to not "work" properly in a system which relies on the face-space.

Algorithm 1. Uniform sampling on the surface of a hyper-ellipsoid

function UNIFORMSAMPLE($V, \partial V, d, S, \epsilon$) ▷ V a d-dimensional hyper-ellipsoid, ∂V: surface of V, S: a uniform sampler on V, ϵ: variation distance from the uniform distribution.
  Draw N points uniformly from V using S,
  Estimate κ – the smallest eigenvalue of the Inertia matrix $E[(\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})^T]$ ▷ $N = O(dlog^2(d)log\frac{1}{\epsilon})$
  $\sqrt{t} \leftarrow \frac{\epsilon\sqrt{\kappa}}{32d}$
  $\boldsymbol{p} \leftarrow \varnothing$
  while $\boldsymbol{p} == \varnothing$ **do**
    $x \in_R S$
    $y \in_R Gaussian(x, 2tI_d)$ ▷ y follows a normal distribution.
    if $y \notin V$ **then**
      $\boldsymbol{p} \leftarrow \overrightarrow{\boldsymbol{xy}} \bigcup \partial V$
    end if
  end while
  return $\boldsymbol{p}$
end function

projection into ∂V is at a distance s away from $\boldsymbol{x}$. We sample a large number of points in ∂V off-line to speed up the search for a sampling direction.

2.4 User Interface of GenFace System

The user interface of GenFace is shown in Fig. 1. The user can set the sampling distance by entering a number into the DISTANCE text box in the PARAMETERS panel. She has options for either sampling offspring faces within, or on the perimeter of a region, centered at the seed face. By pressing the GENERATE button in the SEED panel, a seed face and its eight offspring faces are generated and displayed in the OFFSPRING panel with the seed face positioned in the center. The distance from each offspring face to the seed will be displayed on the top of each offspring face.[6] The user can select any face and save the face image and coefficients by pressing the SAVE AS button. Alternatively, the user can generate a large number of faces using a batch mode: The user may input the number of seed faces and the number of offspring in the BATCH panel. By pressing GENERATE button in the panel, all faces will be generated and saved automatically into a set of sequentially named folders with each folder containing a seed and its offsprings. A progress bar will be shown until all faces have been generated. If the generation procedure is interrupted, the user can resume it by specifying the starting number of the seed face and pressing the GENERATE button. The user is allowed to load saved data into the system and the face image will be displayed in the center of the OFFSPRING panel.

Version 1.0.0 of GenFace is implemented in Matlab and compiled for Windows. Generated images are saved as JPEG files and the corresponding meta

[6] This feature is more relevant to the "sample within" option, as the distance from each offspring image to the seed could be different.

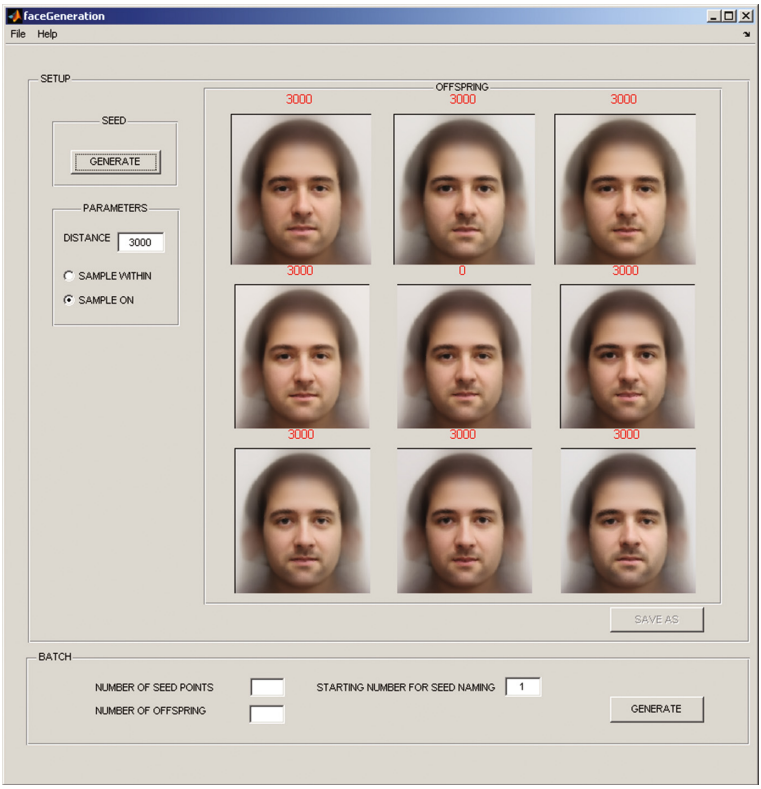


Fig. 1. The user interface of the GenFace system

Algorithm 2. Sampling an offspring face

function SAMPLEOFFSPRING($V, \partial V, d, \boldsymbol{x}, s, \epsilon, k, M$) ▷ V: a d-dimensional hyper-ellipsoidal body, ∂V: the surface of V, $\boldsymbol{x}$: a seed face, s: a sampling distance, ϵ: relative error tolerance of sampling distance, k: the normalized radius of V, M: a diagonal matrix defined as $diag(\sigma_1, \sigma_2, \cdots, \sigma_d)$
  repeat
    Sample a random face $\hat{\boldsymbol{x}}$ uniformly on ∂V
  until $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\| \geq s$
  $t \leftarrow s$, $\hat{t} \leftarrow 0$
  repeat
    $\boldsymbol{p} \leftarrow \boldsymbol{x} + t\boldsymbol{v}$
    $\hat{\boldsymbol{p}} = \frac{k}{\|M^{-1}\boldsymbol{p}\|}\boldsymbol{p}$ ▷ Project $\boldsymbol{p}$ onto ∂V
    $\hat{t} \leftarrow \|\hat{\boldsymbol{p}} - \boldsymbol{x}\|$
    $t \leftarrow \frac{s}{\hat{t}}t$
  until $|\frac{\hat{t}}{s} - 1| \leq \epsilon$
  return $\hat{\boldsymbol{p}}$
end function

data is saved in a text file with the same name. The installation package of the GenFace is publicly available at https://crypto.cs.haifa.ac.il/GenFace/.[7]

3 Testing the GenFace System with SecureFace

In this section we show an example of using synthetic facial images generated by GenFace for evaluating the recently introduced SecureFace system [10] for key derivation from facial images. We start with a brief description of the SecureFace system and then turn to a comparison of its results on real and synthetic data sets of the same size. We also show that using GenFace we can generate data with different levels of fuzziness (simulating the inherent fuzziness of biometric samples) and that this fuzziness directly affects the success rate of the SecureFace system. A similar procedure can be used to choose the parameters for face generation to fit the expected level of fuzziness in the real data. Finally, we test the scalability of the SecureFace system using a much larger set of synthetic faces with the fuzziness parameters that approximate the variation in the real data used in our experiments.

3.1 The SecureFace System

The SecureFace system [10] derives high-entropy cryptographic keys from frontal facial images while fully protecting the privacy of the biometric data, including the training phase. SecureFace (as most of the biometric cryptosystems) offers a two-stage template generation process. Before this process, an input face is aligned to the canonical pose by applying an affine transformation on physical landmarks found by a landmark detector (e.g., [15]).

The first stage of SecureFace converts input images into real-valued representations that suppress the fuzziness in images of the same person due to viewing conditions (such as pose, illumination, camera noise, small deformations etc.). A very large volume of work exists on this topic in the computer vision community. The SecureFace system uses a combination of standard local features that do not require user-dependent training, specifically, *Local Binary Patterns* (LBPs), *Histogram of Oriented Gradients* (HoG), and *Scale Invariant Feature Transform* (SIFT). The extracted features are reduced in dimensionality (by PCA), then concatenated and whitened to remove correlations.

The second phase of the processing transforms real-valued representations (obtained by the first step) to binary strings with the following properties: consistency/discriminability, efficiency (high entropy), and zero privacy loss.

Let $x \in R^D$ be a data point (real-valued feature vector) and w_k be a projection vector. The transformation to a binary string is done by computing $1/2(\mathbf{sgn}(w_k^T x) + 1)$. Vectors w_k form the embedding matrix $W = [w_1, \ldots, w_K]$. W is obtained by a learning process with the goal of mapping the templates

[7] GenFace does not require full Matlab, but the installation package will install the "MATLAB Component Runtime".

of the same person close to each other in the Hamming space and templates of different people far from each other. The resulting binary strings have almost full entropy, specifically, each bit has $\approx 50\%$ chance of being one or zero, and different bits are independent of each other. The embedding, W, is learned on a different (public) set of people, thus it does not reveal any information regarding system's users. Finally, W is generated only once and can be used with any data set of faces without any re-training.

In the acquisition stage a generated template, $t = 1/2(\mathbf{sgn}(W^T x)+1)$, is used as the input to the fuzzy commitment scheme [14] which chooses a codeword C and computes $s = C + t$. The helper data s is released to the user. For a re-sampling, a new image of a user is converted to a binary string using the two-stage template generation process (described above) resulting in a template $t' = 1/2(\mathbf{sgn}(W^T x) + 1)$. The user supplies s and along with t' the system computes $C' = EC(s+t')$, where $EC()$ is the error-correction function. If x and x' are close (in terms of Hamming distance), then $C' = C$.

3.2 Experiments

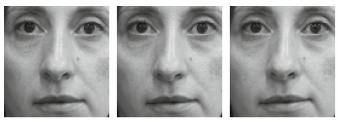
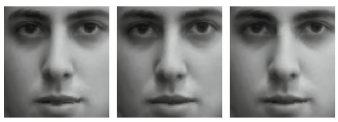
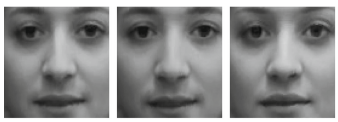
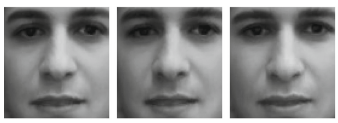
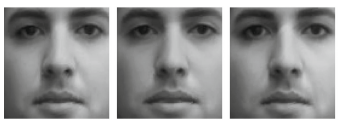
Our experiments include two parts. In the first part, we compare the performance of the SecureFace system on real and synthetic sets of equal sizes, varying the level of fuzziness in the synthetics data. This experiment allows us to choose (if possible) the level of fuzziness that fits the real data. In the second part we test the scalability of the SecureFace system on larger sets of synthetic data using the selected parameters. Acquiring real facial data of that size (5,000–25,000 subjects) is problematic as discussed earlier.

Matching Size Experiment: We used a subset of the in-house dataset of real people, which contains 508 subjects, with 2.36 images per subject on average. All images were collected in the same room while the participants were sitting at the same distance from the camera. The lighting conditions were kept the same during collection, and the subjects were told to keep a neutral expression. The top row of Table 1 illustrates two subjects with 3 images per each from the dataset. Even though, the variation in viewing conditions in this set were kept to a minimum, some variation in lighting (due to the different height of people), pose, and facial expressions is still present. We choose this particular set of real faces in our experiments, as it was used as a test-bed for the SecureFace system and because it focuses on "natural" variation in identity (as opposed to mix of all variations) which the proposed GenFace can offer.

For the synthetic datasets, we generated 5 sets using different parameterization of GenFace. Each set contained 500 seeds with 5 offsprings per seed, which can be viewed as 500 subjects with 5 images per subject. Rows 2–5 of Table 1 demonstrate examples of generated images for different distance parameters (each row corresponds to a different distance parameter and shows two subjects with 3 images per each).

We ran the acquisition stage of the SecureFace system for all 508 subjects. Then we simulated 1,200 genuine attempts and 1,665 imposter attempts (all

Table 1. Example of images from the real set (first row) and from the synthetic set with different levels of fuzziness. Each row shows two different subjects with 3 images per subject.

Real Data	
Synthetic, $d = 2,000$	
Synthetic, $d = 3,000$	
Synthetic, $d = 3,200$	
Synthetic, $d = 3,300$	
Synthetic, $d = 3,500$	

queries in the genuine attempts were with 5 reference subjects chosen uniformly at random from 508). The corresponding ROC curve is depicted in Fig. 2. We note that in the context of SecureFace, a False Positive happens when the face of an imposter "succeeds" to unlock the key of a real user.

Our hypothesis is that the distance parameter controls the fuzziness of the generated data. To test this hypothesis, we tested GenFace on each set, using all generated subjects along with 2,500 genuine and 2,500 imposter attempts (again, all queries in the genuine attempts were with 5 reference subjects chosen uniformly at random from 500). The ROC curves of these experiments are shown in Fig. 2, showing that data sets with higher distances between the offsprings to the corresponding seed result in worse ROC curves. This can happen either due to the increased fuzziness in the images of the same seed (subject) or due to decreased distances between different seeds. We compared the distributions of distances among different subjects between all tested sets and found them well

aligned with each other. Thus, we can conclude that the fuzziness of the same person indeed grows with the distance parameter.

According to the plot, the ROC curve corresponding to the distance of 3,000 is the closest to the ROC of the real data. Thus it can be used for testing the system in the large-scale experiment. The third row of Fig. 2 shows examples of facial images for this distance.

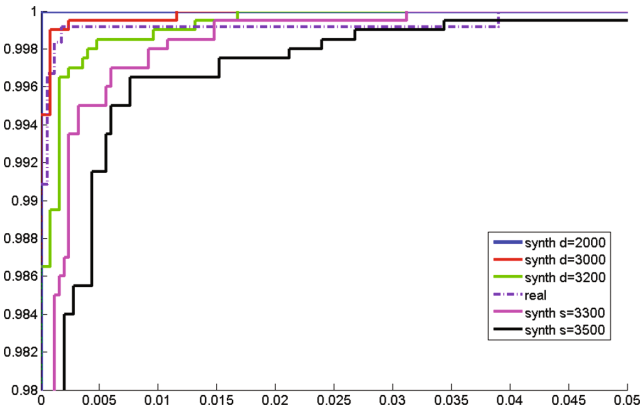


Fig. 2. ROC curves for the "Matching Size" experiment showing the performance SecureFace on real data and on three generated sets with different distance parameters (corresponding to levels of fuzziness among images of the same identity). This figure is best viewed in color. (Color figure online)

Large-Scale Experiment: Testing the scalability of SecureFace (or any other biometric system) is very straightforward with GenFace. Only a small number of parameters need to be set: the distance to the value that approximates the variation in the real data, the number of seeds (subjects), and the number of offspring (images per subject). After that, the required data can be generated in a relatively short time. For example, generating a seed face on a 32-bit Windows XP with an Intel Core i7, 2.67 Hz, 3.24 GB RAM using a code written and compiled in Matlab R2007b (7.5.0.342) takes 0.9 s on average, and generating an offspring for a given seed takes 0.636 s on average. The process can be further optimized and adjusted to run in parallel.

The results of testing the SecureFace using generated sets of different sizes (from 500 to 25,000 subjects) with the distance parameter of 3,000 are shown in Fig. 3. The ROC curves show very plausible behavior that we believe closely approximates that of real data with a similar level of fuzziness.

4 Conclusions

We proposed a system for generating plausible images of faces belonging to the same and different identities. We showed that our system could easily generate a

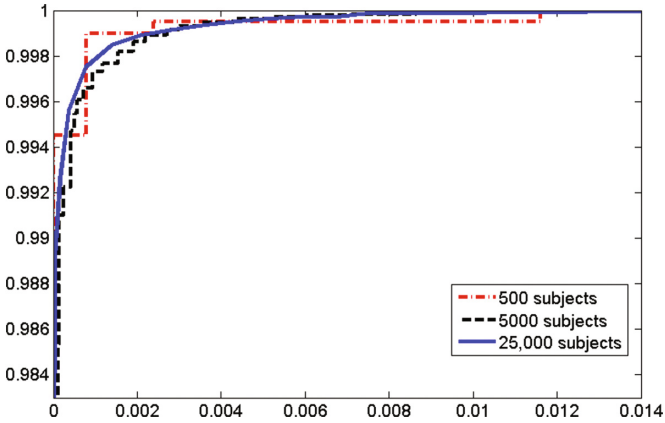


Fig. 3. ROC curves for the "Large-Scale" experiment, showing the performance of SecureFace on synthetic data of different sizes (from 500 to 25,000 subjects) and with a distance of 3,000 (between offsprings and the corresponding seed). The ROC curve corresponding to 500 subjects (d = 3,000) is copied from Fig. 2 for the reference. This figure is best viewed in color. (Color figure online)

very large number of facial images while controlling their fuzziness. We suggested that our system can be used for systematic evaluation of biometric systems, instead of real faces. We showed the merit of this approach using the recently introduced SecureFace.

Future research directions include: Evaluating the merits of using GenFace's images for training purposes. Evaluating facial landmark detectors by comparing their results with GenFace's output (that can contain landmarks). This aspect can be used for improving future landmark detectors, which have various applications throughout computer vision beyond biometrics. In the current work, we addressed only the "natural variation" in facial appearance. Future work will include integrating variations resulting from a change in viewing conditions. This will require rendering facial appearances, using a face 3D model (e.g., such as in [22]). Additionally, studying the difference in the behavior of other biometric systems (e.g., Amazon cognitive systems) when using synthetic faces rather than real ones is left to future works.

Finally, we plan to extend the platform support, e.g., introduce Linux support. We also plan on offering a parallel process for the sampling of faces and offsprings.

Acknowledgements. This research was supported by UK Engineering and Physical Sciences Research Council project EP/M013375/1 and by the Israeli Ministry of Science and Technology project 3-11858. We thank Mahmood Sharif for his support in experiments using SecureFace. We thank the anonymous reviewers of this paper for their ideas and suggestions.

References

1. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 711–720 (1997)
2. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pp. 187–194. ACM Press/Addison-Wesley Publishing Co. (1999)
3. Bringer, J., Chabanne, H., Favre, M., Patey, A., Schneider, T., Zohner, M.: GSHADE: faster privacy-preserving distance computation and biometric identification. In: Unterweger, A., Uhl, A., Katzenbeisser, S., Kwitt, R., Piva, A. (eds.) ACM Information Hiding and Multimedia Security Workshop, IH&MMSec 2014, Salzburg, Austria, June 11–13, 2014, pp. 187–198. ACM (2014)
4. Cappelli, R., Maio, D., Maltoni, D.: SFinGe: an approach to synthetic fingerprint generation. In: International Workshop on Biometric Technologies, pp. 147–154 (2004)
5. Chang, Y., Zhang, W., Chen, T.: Biometrics-based cryptographic key generation. In: IEEE International Conference on Multimedia and Expo (ICME), pp. 2203–2206 (2004)
6. Chen, C., Veldhuis, R., Kevenaar, T., Akkermans, A.: Biometric binary string generation with detection rate optimized bit allocation. In: CVPR Workshop on Biometrics, pp. 1–7 (2008)
7. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. IEEE Trans. Pattern Anal. Mach. Intell. **23**(6), 681–685 (2001)
8. Dezert, J., Musso, C.: An efficient method for generating points uniformly distributed in hyperellipsoids. In: The Workshop on Estimation, Tracking and Fusion: A Tribute to Yaakov Bar-Shalom (2001)
9. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.D.: Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. SIAM J. Comput. **38**(1), 97–139 (2008)
10. Dunkelman, O., Osadchy, M., Sharif, M.: Secure authentication from facial attributes with no privacy loss. In: Sadeghi, A., Gligor, V.D., Yung, M. (eds.) 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS 2013, Berlin, Germany, November 4–8, 2013, pp. 1403–1406. ACM (2013)
11. Edwards, G.J., Cootes, T.F., Taylor, C.J.: Face recognition using active appearance models. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 581–595. Springer, Heidelberg (1998). doi:10.1007/BFb0054766
12. Gibson, S.J., Solomon, C.J., Bejarano, A.P.: Synthesis of photographic quality facial composites using evolutionary algorithms. In: Proceedings on British Machine Vision Conference, BMVC 2003, Norwich, UK, pp. 1–10, September 2003 (2003)
13. Gibson, S.J., Solomon, C.J., Pallares-Bejarano, A.: Nonlinear, near photorealisticcaricatures using a parametric facial appearance model. Behav. Res. Methods **37**(1), 170–181 (2005). http://dx.doi.org/10.3758/BF03206412
14. Juels, A., Wattenberg, M.: A fuzzy commitment scheme. In: Motiwalla, J., Tsudik, G. (eds.) CCS 1999, Proceedings of the 6th ACM Conference on Computer and Communications Security, Singapore, November 1–4, 1999, pp. 28–36. ACM (1999)
15. Kazemi, V., Sullivan, J.: One millisecond face alignment with an ensemble of regression trees. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23–28, 2014, pp. 1867–1874 (2014)

16. Lee, K., Byatt, G., Rhodes, G.: Caricature effects, distinctiveness, and identification: testing the face-space framework. Psychol. Sci. **11**(5), 379–385 (2000)
17. Lewis, M.: Face-space-R: towards a unified account of face recognition. Vis. Cogn. **11**(1), 29–69 (2004)
18. Li, M., Zuo, W., Zhang, D.: Convolutional network for attribute-driven and identity-preserving human face generation. arXiv preprint arXiv:1608.06434 (2016)
19. Masi, I., Tran, A.T., Hassner, T., Leksut, J.T., Medioni, G.: Do we really need to collect millions of faces for effective face recognition? In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 579–596. Springer, Cham (2016). doi:10.1007/978-3-319-46454-1_35
20. Matthews, I., Baker, S.: Active appearance models revisited. Int. J. Comput. Vis. **60**(2), 135–164 (2004)
21. Narayanan, H., Niyogi, P.: Sampling hypersurfaces through diffusion. In: Goel, A., Jansen, K., Rolim, J.D.P., Rubinfeld, R. (eds.) APPROX/RANDOM - 2008. LNCS, vol. 5171, pp. 535–548. Springer, Heidelberg (2008). doi:10.1007/978-3-540-85363-3_42
22. Paysan, P., Knothe, R., Amberg, B., Romdhani, S., Vetter, T.: A 3D face model for pose and illumination invariant face recognition. In: Tubaro, S., Dugelay, J. (eds.) Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2009, 2–4 September 2009, Genova, Italy, pp. 296–301. IEEE Computer Society (2009).,
23. Solomon, C.J., Gibson, S.J., Mist, J.J.: Interactive evolutionary generation of facial composites for locating suspects in criminal investigations. Appl. Soft Comput. **13**(7), 3298–3306 (2013)
24. Sumi, K., Liu, C., Matsuyama, T.: Study on synthetic face database for performance evaluation. In: Zhang, D., Jain, A.K. (eds.) ICB 2006. LNCS, vol. 3832, pp. 598–604. Springer, Heidelberg (2005). doi:10.1007/11608288_79
25. Turk, M., Pentland, A.: Eigenfaces for recognition. J. Cogn. Neurosci. **3**(1), 71–86 (1991)
26. Tuyls, P., Akkermans, A.H.M., Kevenaar, T.A.M., Schrijen, G.-J., Bazen, A.M., Veldhuis, R.N.J.: Practical biometric authentication with template protection. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 436–446. Springer, Heidelberg (2005). doi:10.1007/11527923_45
27. Tzimiropoulos, G., Pantic, M.: Optimization problems for fast AAM fitting in-the-wild. In: IEEE International Conference on Computer Vision, ICCV, pp. 593–600 (2013)
28. Valentine, T.: A unified account of the effects of distinctiveness, inversion, and race in face recognition. Q. J. Exp. Psychol. **43**(2), 161–204 (1991)
29. Wu, H., Liu, X., Doretto, G.: Face alignment via boosted ranking model. In: 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24–26 June 2008, Anchorage, Alaska, USA (2008)
30. Zhang, L., Lin, L., Wu, X., Ding, S., Zhang, L.: End-to-end photo-sketch generation via fully convolutional representation learning. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, pp. 627–634. ACM (2015)

http://www.springer.com/978-3-319-60079-6

Cyber Security Cryptography and Machine Learning
First International Conference, CSCML 2017,
Beer-Sheva, Israel, June 29-30, 2017, Proceedings
Dolev, S.; Lodha, S. (Eds.)
2017, XII, 307 p. 59 illus., Softcover
ISBN: 978-3-319-60079-6