IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier XX.XXXX/ACCESS.2019.DOI

A privacy-preserving filter for oblique face images based on adaptive hopping Gaussian mixtures

OMAIR SARWAR[1,2], BERNHARD RINNER[1], AND ANDREA CAVALLARO [2]
[1]Institute of Networked and Embedded Systems, University of Klagenfurt, Austria (e-mail: omair_sarwar@hotmail.com, bernhard.rinner@aau.at)
[2]Centre for Intelligent Sensing, Queen Mary University of London, UK (e-mail: a.cavallaro@qmul.ac.uk)

Corresponding author: Omair Sarwar (e-mail: omair_sarwar50@hotmail.com).

ABSTRACT Photographs taken in public places often contain faces of bystanders thus leading to a perceived or actual violation of privacy. To address this issue, we propose to pseudo-randomly modify the appearance of face regions in the images using a privacy filter that prevents a human or a face recogniser from inferring the identity of people. The filter, which is applied only when the resolution is high enough for a face to be recognisable, adaptively distorts the face appearance as a function of its resolution. Moreover, the proposed filter locally changes the values of its parameters to counter attacks that attempt to estimate them. The filter exploits both global adaptiveness to reduce distortion and local parameter hopping to make their estimation difficult for an attacker. In order to evaluate the efficiency of the proposed approach, we consider an important scenario of oblique face images: photographs taken with low altitude Micro Aerial Vehicles (MAVs). We use a state-of-the-art face recognition algorithm and synthetically generated face data with 3D geometric image transformations that mimic faces captured from an MAV at different heights and pitch angles. Experimental results show that the proposed filter protects privacy while reducing distortion, and is also robust against attacks.

INDEX TERMS Image privacy protection, hopping Gaussian blur, micro aerial vehicles

I. INTRODUCTION

Photography in public places raises privacy concerns as bystanders who happen to be within the field of view of a camera are captured as well. The identity of bystanders could be protected by locating and removing (or sufficiently distorting) key image regions, such as faces, using privacy filters. However, in order to maintain the aesthetic value of an image, only a minimal distortion should be introduced in an image to preserve its content.

A privacy filter for photography should satisfy the following properties: (a) introduce only a minimal distortion; (b) be robust against attacks; and (c) be computationally efficient. Minimal distortion is necessary to maintain the quality of a protected image close to the unprotected one so that the attention of a viewer is not diverted. Therefore blanking out a face [1] is not a desirable option. Robustness is important to avoid privacy violations by various attacks, e.g. brute-force, naïve, parrot and reconstruction attacks [2–12]. A brute-force attack tries to decipher the protected probe images by an exhaustive search [3, 5]. Other attacks use gallery images in addition to the protected probe images [4, 6–8]. In a naïve attack, the protected probe images are compared against the unprotected gallery images [4, 6, 7]. In a parrot attack, the attacker has knowledge about the privacy filter and can transform the gallery images into the distorted domain [4]. In a reconstruction attack, the attacker has some knowledge of how to (partially) reconstruct the probe image from the protected to the unprotected domain [2]. Examples of reconstruction methods include inverse filtering and super-resolution techniques [2, 8]. Finally, computational efficiency is desirable when the filter operates under limited computational and battery power, such as for example in the case of an MAV.

Recent frameworks that support facial privacy-preservation in airborne cameras are Generic Data Encryption [13], Unmanned Aircraft Systems-Visual Privacy Guard [14] and Adaptive Gaussian Blur [15]. Generic Data Encryption sends an encrypted face region to a privacy server that Gaussian blurs or mosaics the face and then forwards it to an end-user. Unmanned Aircraft Systems-Visual Privacy Guard [14] and

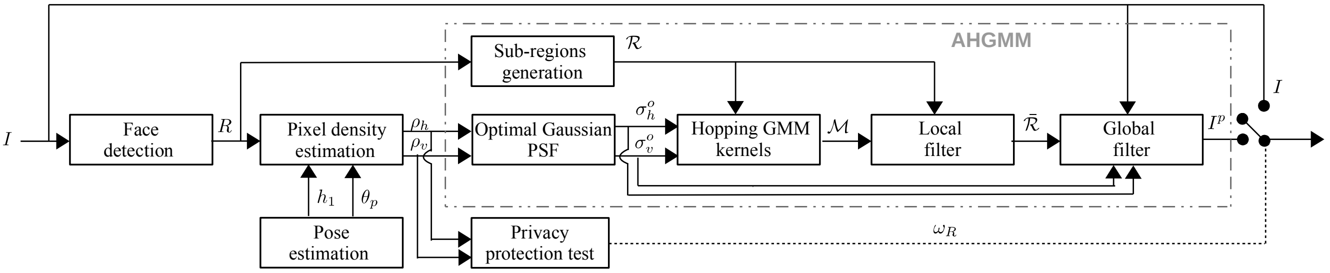


FIGURE 1: Block diagram of the proposed Adaptive Hopping Gaussian Mixture Model (AHGMM) filter. KEY – ρ_h, ρ_v: number of pixels (px) per unit distance (cm) (pixel densities) of a sensitive region R; h_1, θ_P: altitude and tilt angle of the camera used to calculate the pixel densities; ω_R: control signal generated from the pixel densities to decide when to protect R; $\mathcal{R}$: sub-regions of R; σ_o^o, σ_v^o: standard deviations for the hopping Gaussian mixture model $\mathcal{M}$ that filters $\mathcal{R}$ to generate the protected sub-regions $\bar{\mathcal{R}}$; I^p: protected image.

Adaptive Gaussian Blur [15] are aimed instead at on-board implementation with the objective to reduce latency and discourage brute-force attacks on the server [13]. Adaptive Gaussian Blur adaptively configures the Gaussian kernel depending upon the face resolution in order to minimise distortion, while Unmanned Aircraft Systems-Visual Privacy Guard blurs faces with a fixed filter. These methods are prone to parrot attacks [4] on the Gaussian blur.

In this paper, we present a novel privacy protection filter to be used on-board an MAV. The proposed filter distorts a face region with secret parameters to be robust to naïve, parrot and reconstruction attacks. The distortion is minimal and adaptive to the resolution of the captured face: we select the smallest Gaussian kernel that reduces the face resolution below a certain threshold. The selected threshold protects the face against the naïve attack as well as maintains its resolution at a specified level. To prevent other attacks, we then insert supplementary Gaussian kernels in the selected Gaussian kernel and hop their parameters locally using a pseudorandom number generator (PRNG) so their estimation from the filtered face image is made difficult. The block diagram of the proposed filter is shown in Figure 1. In summary, the main contributions of this paper are: (1) the idea of using Gaussian hopping kernels for privacy and utility preservation, (2) the generation of a large-scale synthetic face image data set emulating faces captured from an MAV, and (3) extensive experiments to validate the proposed Gaussian hopping kernels, including reconstruction attacks.

The paper is organised as follows. Sec. II covers the state-of-the-art in visual privacy protection filters. Sec. III defines the problem. Sec. IV describes the proposed algorithm, and discusses its computational complexity and security level. Sec. V presents our face data set generation and Sec. VI discuss the experimental results. Finally, Sec. VII concludes the paper.

II. BACKGROUND

Visual privacy protection filters can be applied as a pre-processing or post-processing step (Fig. 2).

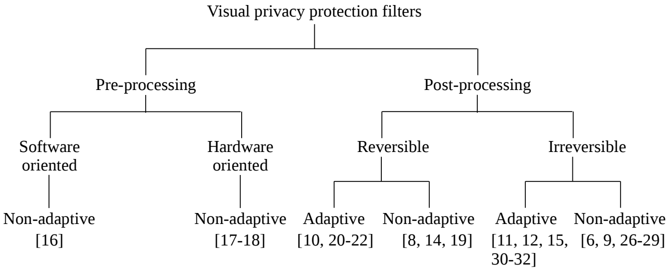


FIGURE 2: A taxonomy of visual privacy protection filters.

Pre-processing privacy filters are irreversible and operate during image acquisition to prevent a camera from capturing sensitive regions. These filters disable the software or hardware of the camera or notify about photography prohibition [18]. Hardware based filters prevent the camera from taking images for example by bursting back an intense light for flash photography [19, 20] or by detecting human face/body using an infrared sensor and then obfuscating using a spatial light modulator sensor placed in front of the Charge Coupled Device (CCD) sensor [21, 22].

Post-processing privacy filters protect sensitive regions after image acquisition and can be reversible or irreversible. Reversible filters conceal sensitive regions using a private key, which can later be used to recover the original sensitive region. Irreversible filters deform the features of a sensitive region permanently. Both reversible and irreversible filters can be non-adaptive or adaptive.

Reversible non-adaptive filters are based on generic encryption [3, 10, 23–25]. Reversible adaptive filters include scrambling [5, 10–12, 26–28], warping [29] and morphing [30]. While reversible adaptive filters are robust against a parrot attack, their protected faces can be compromised by spatial-domain [31, 32] or frequency-domain attacks [33].

Irreversible non-adaptive filters blank out [1, 34] or replace a face with a de-identified representation [4]. For example, to maintain k-anonymity, the algorithm k-Same [4] replaces k faces with their average face. Variants of this algorithm

TABLE 1: Post-processing privacy filters. KEY – DCT-S: Discrete Cosine Transform Scrambling [5]; PICO: Privacy through Invertible Cryptographic Obscuration [3]; GARP: Gender, Age and Race Preservation [16]; UAS-VPG: Unmanned Aircraft Systems-Visual Privacy Guard [14]; Cartooning [6]; SVGB: Space Variant Gaussian Blur [17]; ODBVP: Optimal Distortion-Based Visual Privacy [7]; AGB: Adaptive Gaussian Blur [15]. Adaptive control modulates the strength of a privacy filter.

			DCT-S	PICO	GARP	UAS-VPG	Cartooning	SVGB	ODBVP	AGB	Proposed
Distortion	adaptive control	image based	✓				✓	✓	✓		
		navigation sensors								✓	✓
	2D kernel	isotropic					✓	✓	✓		
		anisotropic								✓	✓
Robustness	to brute-force attack				✓	✓	✓	✓	✓	✓	✓
	to naïve attack		✓	✓	✓	✓	✓	✓	✓	✓	✓
	to inverse filter attack				✓		✓				✓
	to super-resolution attack		✓	✓	✓						✓
	to parrot attack	with detectors			✓						
		without detectors	✓	✓							✓
Computational simplicity						✓		✓	✓	✓	

use additional specialised detectors to then preserve attributes such as facial expressions, pose, gender, race and age [16, 35–37]. Neural network based filters (e.g. Generative Adversarial Networks (GANs)) also preserve such attributes using prior knowledge [38–45].

Irreversible non-adaptive filters are robust to parrot attacks. Irreversible adaptive filters lower the resolution of a sensitive region so that humans or algorithms cannot recognise the identity. Examples include pixelation [9, 46], Gaussian blur [47] and cartooning [6]. The kernel size of the privacy filters can be manually selected [6, 7] or only the centre kernel size is manually selected and then the Space Variant Gaussian Blur (SVBG) filter [17] automatically decreases the kernel size from the centre to the boundary of the detected face. Adaptive Gaussian Blur (AGB) [15] exploits the different horizontal and vertical resolutions that are typical in aerial photography and automatically adapts an anisotropic kernel based on the resolution of the detected face. However, irreversible adaptive filters are vulnerable to parrot attacks.

As a summary, Table 1 compares representative filters for the following categories: reversible & adaptive [5], reversible & non-adaptive [3], and irreversible & non-adaptive filters [16]. The remaining [6, 7, 14, 15, 17] and our proposed approaches are irreversible & adaptive filters.

III. PROBLEM DEFINITION

Let the set $\mathcal{D} = \{\mathcal{R}_k\}_{k=1}^{K}$ contain face data of K subjects, where k represents the identity of a person (label). Let each subject k appear in at most Z images, i.e. $\mathcal{R}_k = \{R_i | i \leq Z\}$. Let $\mathcal{R}_\mathcal{G}, \mathcal{R}_\mathcal{P} \subset \mathcal{D}$ be the gallery and probe sets, respectively. Usually $|\mathcal{R}_\mathcal{G}| > |\mathcal{R}_\mathcal{P}|$, where $|.|$ is the cardinality of a set, and $\mathcal{R}_\mathcal{G} \cap \mathcal{R}_\mathcal{P} = \emptyset$.

A. PRIVACY FILTER

Let a privacy filter $F_{\Omega_j} : \mathcal{R}_\mathcal{P} \rightarrow \bar{\mathcal{R}}_\mathcal{P}$ distort image features in order to reduce the probability P for an attacker to correctly predict labels. This operation produces a protected probe set $\bar{\mathcal{R}}_\mathcal{P}$, whose distortion depends on Ω_j, where $j \in \{h, v\}$ indicates the horizontal and vertical direction in an image. Let the distortion generated by F_{Ω_j} be measured by the Peak Signal to Noise Ratio (PSNR):

$$PSNR = 20 \log_{10} \frac{R_{max}}{\sqrt{MSE}}, \tag{1}$$

where R_{max} is the dynamic range of the pixel values. The mean square error, MSE, between the pixel intensities of an unprotected, $R \in \mathcal{R}_\mathcal{P}$, and protected, $\bar{R} \in \bar{\mathcal{R}}_\mathcal{P}$, face is

$$MSE = \frac{1}{|\mathcal{R}_\mathcal{P}|WH} \sum_{r=1}^{|\mathcal{R}_\mathcal{P}|} \sum_{w=1}^{W} \sum_{h=1}^{H} ||R(w,h) - \bar{R}(w,h)||_r^2, \tag{2}$$

where W and H are the width and height of R, respectively.

We relate the privacy level of a face region to the accuracy η of a face recogniser [6, 7]. The value of η is the commutative rank-n for face identification or the Equal Error rate (EER) for face verification. We consider in this paper face verification, thus

$$\eta = \frac{TP + TN}{|\mathcal{R}_\mathcal{P}|}, \tag{3}$$

where TP and TN are true positives and true negatives, respectively. Our target is to force a face recogniser of an attacker to have the accuracy of a random classifier, which for face verification is $\epsilon = 0.5$.

We therefore aim to design F_{Ω_j} that irreversibly but minimally distorts the appearance of R so that the identity is not recognisable with a probability higher than a random guess.

If $E(w,h) = F_{\Omega_j}(R(w,h)) - R(w,h)$, the ideal distortion parameter, Ω_j^o, should be derived as:

$$\Omega_j^o = \arg\min_{\Omega_j} \left(\frac{1}{WH} \sum_{w=1}^{W} \sum_{h=1}^{H} E(w,h) + (P(F_{\Omega_j}(R)|\mathcal{B}) - 0.5) \right), \tag{4}$$

where $\mathcal{B} \in \{\mathcal{R}_\mathcal{G}, \bar{\mathcal{R}}_\mathcal{G}, \hat{\mathcal{R}}_\mathcal{G}\}$. The first term aims to introduce a minimal distortion, whereas the second term forces the classification results to be equivalent to those of a random classifier, irrespective of whether the filtered or reconstructed face is compared against the unprotected, filtered or reconstructed gallery data sets. The second term is dependent upon the recognition capability of a face recogniser and is heuristically calculated for a given face recogniser, i.e. the minimum distortion at which the face recogniser starts behaving similarly to a random classifier [48–50].

B. ATTACKS

The content of R should be protected against naïve-T, parrot-T and reconstruction attacks. Let an attacker have access to $\mathcal{B} \in \{\mathcal{R}_\mathcal{G}, \bar{\mathcal{R}}_\mathcal{G}, \hat{\mathcal{R}}_\mathcal{G}\}$, where $\bar{\mathcal{R}}_\mathcal{G}$ is the filtered gallery data set and $\hat{\mathcal{R}}_\mathcal{G}$ is the filtered and reconstructed gallery data set. An attacker can modify $\bar{\mathcal{R}}_\mathcal{P}$, $\mathcal{R}_\mathcal{G}$, or both, to correctly predict $\tilde{K}$ of $\bar{\mathcal{R}}_\mathcal{P}$. In a naïve attack (here referred to as naïve-T attack), a privacy filter is applied on $\mathcal{R}_\mathcal{P}$ to generate a protected probe data set $\bar{\mathcal{R}}_\mathcal{P}$, while the unaltered $\mathcal{R}_\mathcal{G}$ is used for training [4]. A parrot attack (here referred to as parrot-T attack), learns the privacy filter type and its parameters Ω_j (e.g. Gaussian blur of certain standard deviation used to generate $\bar{\mathcal{R}}_\mathcal{P}$). Then, the learned filter is applied on $\mathcal{R}_\mathcal{G}$ to generate a privacy protected gallery data set $\bar{\mathcal{R}}_\mathcal{G}$. Finally, $\bar{\mathcal{R}}_\mathcal{G}$ and $\bar{\mathcal{R}}_\mathcal{P}$ are used for training and testing, respectively [4]. In a reconstruction attack, the discriminating features of $\bar{\mathcal{R}}_\mathcal{P}$ are first restored (e.g. using an inverse filter or a super-resolution algorithm) to generate a reconstructed probe data set $\hat{\mathcal{R}}_\mathcal{P}$ and then compared against $\mathcal{R}_\mathcal{G}$ or a reconstructed gallery data set $\hat{\mathcal{R}}_\mathcal{G}$. An inverse filter first estimates the parameters of a privacy filter using $\bar{\mathcal{R}}_\mathcal{P}$ and then performs an inverse operation to reconstruct the original faces [2]. Similarly, a super-resolution algorithm first learns embeddings (i.e. relationships) between the high-resolution and their corresponding low-resolution faces and then reconstructs the high-resolution faces for $\bar{\mathcal{R}}_\mathcal{P}$ [8].

IV. PROPOSED APPROACH

In order to minimally distort R as well as to achieve robustness against brute-force, naïve-T, parrot-T and reconstruction attacks, we propose the Adaptive Hopping Gaussian Mixture Model (AHGMM) algorithm. AHGMM consists of a globally estimated optimal Gaussian Point Spread Function (PSF) and supplementary Gaussian PSFs added inside the optimal Gaussian PSF. For a single supplementary Gaussian PSF inside an optimal Gaussian PSF, AHGMM is illustrated in Fig. 3, while the pseudo-code is given in Algorithm 1. (A list with the notation used in this paper is presented in Table 4.)

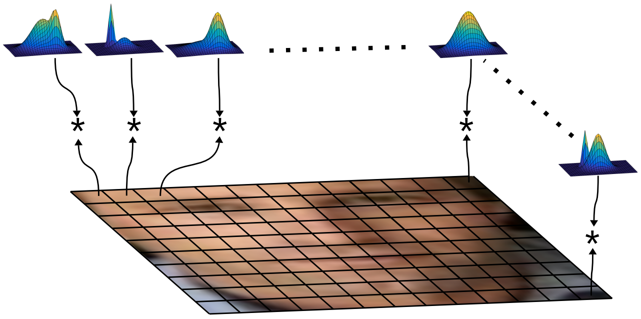


FIGURE 3: Visualisation of local filtering in AHGMM. The face region R is divided into N sub-regions and each sub-region R_n is convolved ($*$) with a hopping Gaussian mixture model kernel M_n, which is made of an optimal Gaussian function and one (or more) supplementary Gaussian function added inside the optimal Gaussian function. While convolving with each sub-region of the face, the optimal and the supplementary Gaussian functions change their parameters, i.e. mean and standard deviation, which in turn change the shape of the Gaussian mixture model based kernel.

A. PIXEL DENSITY ESTIMATION

Let an MAV capture an image I while flying at an altitude of h_1 meters. Let the principal axis $\boldsymbol{P}$ of its on-board camera be tilted by θ_P from the nadir direction $\boldsymbol{N}$ (see Figure 4). We assume that height h_1 and tilt angle θ_P of the camera can be estimated.

A value of $\theta_P \neq 0$ generates an oblique image. Let h_2 be the height of the face above ground[1]. We represent the face region in the image as $R \in \mathcal{R}_k \subset \mathcal{D}$, which is viewed at an angle θ_R.

Let ρ_j represent the pixel density (px/cm) around the centre C_R of R. If p_h and p_v represent the physical dimensions of a pixel in the horizontal and vertical direction, respectively and f is the focal length of the camera, the horizontal density ρ_h for a pixel around C_R [15] is

$$\rho_h = \frac{f cos(\theta_R)}{p_h(h_1 - h_2)}, \tag{5}$$

and the vertical density ρ_v, by exploiting the small angle approximation for a single pixel of the image sensor [15], is

$$\rho_v \approx \frac{f cos(\theta_R) sin(\theta_R)}{p_v(h_1 - h_2)}. \tag{6}$$

Let $\omega_R \in \{0,1\}$ define whether R is naturally protected ($\omega_R = 0$) because of a low horizontal and vertical density, or not ($\omega_R = 1$) [15]:

$$\omega_R = \begin{cases} 1 & if \quad \rho_h > \rho_h^o \quad \text{and} \quad \rho_v > \rho_v^o \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where ρ_h^o and ρ_v^o are pixel densities at which a state-of-the-art machine algorithm starts recognising human faces, and

[1] While each image I could contain L faces, for simplicity we consider in this paper only the case $L = 1$.

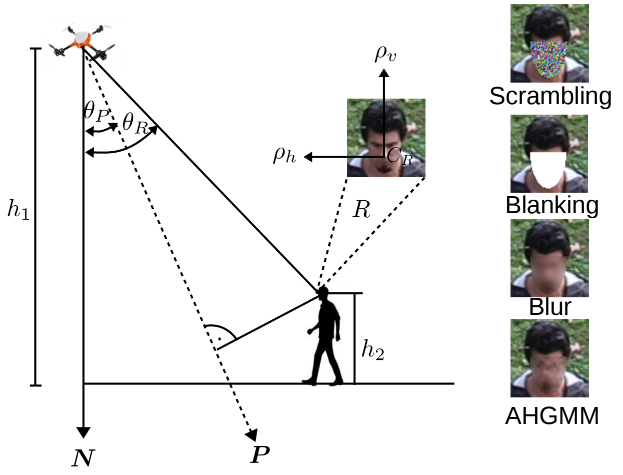


FIGURE 4: Capturing an image with an airborne camera at height h_1. The principal axis $\boldsymbol{P}$ of the camera is tilted by θ_P from the nadir direction $\boldsymbol{N}$. The face region R, at height h_2 above the ground, is viewed at an angle θ_R. The variables ρ_h and ρ_v represent the horizontal and vertical pixel density of R at its centre C_R in the captured image. Four sample images show a scrambled, blanked, Gaussian blurred and AHGMM filtered image, which is captured at $\theta_P = \theta_R = 50°$.

simply called thresholds. If $\omega_R = 0$, then the original frame I can be transmitted without any modifications. Otherwise, R should be protected by a privacy filter to reduce its pixel densities below ρ_h^o and ρ_v^o. When R is not inherently protected, we assume that the corresponding bounding box is given.

B. OPTIMAL GAUSSIAN PSF

A 2D PSF $g(h,v)$, or impulse response, is the output of a filter when the input is a point source. In the discrete domain [51], it is given as $g(h,v) = \delta(h,v) * g(h,v)$, where $*$ is the convolution operation and

$$\delta(h,v) = \begin{cases} 1 & \text{if } h = v = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

In the case of Gaussian blur, $g(h,v)$ is an approximated Gaussian function of mean $\mu_j = 0$ and standard deviation $\sigma_j > 0$ [7, 15, 17], and thus called a Gaussian PSF of parameter $\Omega_j = (\mu_j, \sigma_j)$. More specifically, the parameter $\sigma_j \in \{\sigma_{jl} | l \in \mathbb{N}, \sigma_{jl+1} > \sigma_{jl}\}$ controls the distortion strength of F_{Ω_j} and provides pixel density $\rho_j \in \{\rho_{jl} | l \in \mathbb{N}, \rho_{jl+1} < \rho_{jl}\}$ in $\bar{R}$, respectively.

As a higher σ_j results in a lower ρ_j, we first find the minimum value called optimal parameter σ_j^o of σ_j that makes $\rho_j < \rho_j^o$. As a result, σ_j^o provides the minimum distortion in $\bar{R}$ while making it robust against the naïve-T attack (i.e. $P(\bar{R}|\mathcal{R}_{\mathcal{G}}) \to \epsilon$). Increasing σ_j beyond σ_j^o increases the distortion without improving the privacy level as the recogniser performance is already at the level of a random classifier. For a face captured from an MAV with pixel densities ρ_j, we calculate $\Omega_j^o = (\mu_j^o, \sigma_j^o)$ of an optimal Gaussian PSF (lines 2-5 in Algorithm 1), where $\mu_j^o = 0$ like in traditional Gaussian blur [7, 15, 17] and σ_j^o [15] is estimated as described below.

A Gaussian PSF of standard deviation σ_j^o in the spatial domain is another Gaussian PSF of standard deviation $\acute{\sigma}_j^o$ in the frequency domain and both the Gaussian PSFs are related as

$$\acute{\sigma}_j^o = \frac{\rho_j}{2\pi\sigma_j^o}, \tag{9}$$

where $\acute{\sigma}_j^o$ is measured in cycles/cm, σ_j^o in px and ρ_j in px/cm. Let f_s represent the Nyquist frequency of ρ_j. Let $f_s^o < f_s$ be the highest spatial frequency component that we want to completely remove using a low-pass filter (i.e. Gaussian blur). We can consider f_s^o to be the Nyquist frequency for ρ_j^o, i.e. the pixel density after filtering, and therefore

$$\rho_j^o = 2f_s^o. \tag{10}$$

As we are interested in removing frequency components beyond f_s^o, we can select $f_s^o = 3\acute{\sigma}_j^o$ because the amplitude response of a Gaussian PSF at three times of its standard deviation is very close to zero and multiplication (convolution in space domain) with such a Gaussian PSF will suppress frequencies larger than f_s^o. Substituting $f_s^o = 3\acute{\sigma}_j^o$ in Eq. 10, in the resulting relation Eq. 9 and finally rearranging gives the optimal standard deviation of Gaussian PSF as

$$\sigma_j^o = \frac{3\rho_j}{\pi\rho_j^o}. \tag{11}$$

C. HOPPING GMM KERNELS

Filtering R with the optimal Gaussian PSF defined by Ω_j^o would only protect R from a naïve-T attack but not from a parrot-T attack and a reconstruction attack. To ensure that the probability of correctly predicting the label of $\bar{R}$ is not increased in case of the parrot-T attack (i.e. $P(\bar{R}|\bar{\mathcal{R}}_{\mathcal{G}}) \to \epsilon$) as well as the reconstruction attack (i.e. $P(\hat{R}|\mathcal{R}_{\mathcal{G}}) \to \epsilon$ or $P(\hat{R}|\hat{\mathcal{R}}_{\mathcal{G}}) \to \epsilon$), we secretly modify Ω_j^o to $\bar{\Omega}_j^o$ while generating $\bar{R}$ so that an adversary is unable to accurately reconstruct face region $\hat{R}$, or even to generate $\hat{\mathcal{R}}_{\mathcal{G}}$ and $\bar{\mathcal{R}}_{\mathcal{G}}$. For this purpose, we generate a set $\mathcal{R}$ which consists of N sub-regions in such a way that each sub-region covers a small area of R:

$$\mathcal{R} = \Big\{ R_n | n \in [1, N] \Big\}. \tag{12}$$

The size of R_n (in pixels) affects the total number of sub-regions N per face region R, which could influence its privacy level. Smaller values of N (larger sub-regions) result in a reduced distortion.

After finding $\Omega_j^o = (\mu_j^o, \sigma_j^o)$ and generating $\mathcal{R}$, we make a hopping mixture of Gaussian for each sub-region, i.e. we pseudo-randomly change Ω_j^o to $\bar{\Omega}_j^o$ for each R_n. Moreover, we select supplementary Gaussian PSFs inside this optimal Gaussian PSF and vary their parameters based on pseudo-random weights (lines 9-17 in Algorithm 1).

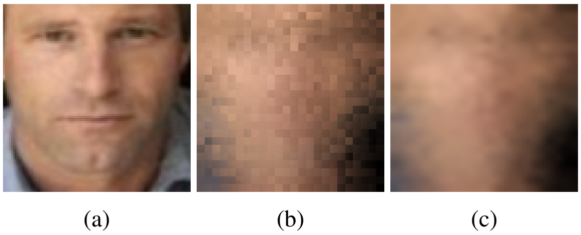


FIGURE 5: Minimising blocking artefacts of spatially hopping Gaussian functions in the AHGMM filter by a convolution with a global kernel. (a) Original image of 96 × 96 pixels from the LFW data set, (b) image after local filtering in AHGMM showing blocking artefacts and (c) image after the local filtering followed by the global filtering in AHGMM.

Let the set $\mathcal{X}$ contain the parameters of the modified optimal and supplementary Gaussian PSFs for each sub-region. This set is represented as

$$\mathcal{X} = \Big\{(\mu_{jm}, \sigma_{jm})_n | n \in [1, N], j \in \{h, v\}, m \in [0, M]\Big\}, \tag{13}$$

where M is the number of supplementary Gaussian PSFs. The element $m = 0$ represents the modified optimal Gaussian PSF given by

$$\mu_{j0} = \pm\alpha_{j0}\sigma_j^o, \tag{14}$$

$$\sigma_{j0} = (1 \pm \beta_{j0})\sigma_j^o, \tag{15}$$

while the remaining elements (i.e. $m \in (0, M]$) belong to the supplementary Gaussian PSFs. These elements are calculated as

$$\mu_{jm} = \pm\alpha_{jm}\sigma_j^o\gamma_{jm}, \tag{16}$$

$$\sigma_{jm} = (1 \pm \beta_{jm})\sigma_j^o\gamma_{jm}, \tag{17}$$

where $\alpha_{jm} \in [0, 1]$ and $\beta_{jm} \in [0, 1]$ are normalised pseudo-randomly generated numbers and control the local distortion in filtering. The variable $\gamma_{jm} \in (0, 1]$ controls the relative size of the supplementary Gaussian PSF w.r.t. the optimal Gaussian PSF.

After generating the parameters of the Gaussian PSFs, a set $\mathcal{G}$ representing 2D anisotropic-discretised Gaussian PSFs corresponding to $\mathcal{X}$ is created as

$$\mathcal{G} = \Big\{G_{nm} | n \in [1, N], m \in [0, M]\Big\}, \tag{18}$$

where each G_{nm} is calculated (line 19 in Algorithm 1) as [52]

$$G_{nm} \approx A_{nm} e^{-\left(\frac{(h-\mu_{hnm})^2}{2\sigma_{hnm}^2} + \frac{(v-\mu_{vnm})^2}{2\sigma_{vnm}^2}\right)}, \tag{19}$$

where

$$A_{nm} = 1 \Bigg/ \sum_{(h,v)\in d} e^{-\left(\frac{(h-\mu_{hnm})^2}{2(\sigma_{hnm})^2} + \frac{(v-\mu_{vnm})^2}{2(\sigma_{vnm})^2}\right)}, \tag{20}$$

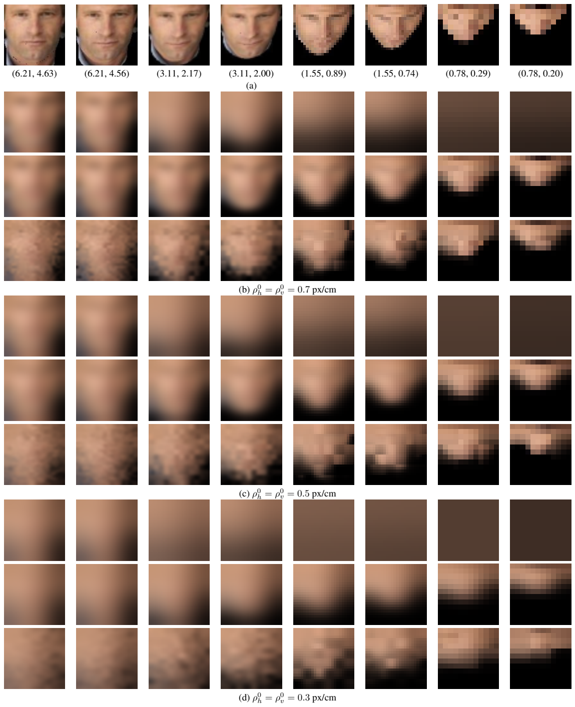

(6.21, 4.63)
(6.21, 4.56)
(3.11, 2.17)
(3.11, 2.00)
(1.55, 0.89)
(1.55, 0.74)
(0.78, 0.29)
(0.78, 0.20)
(a)
(b) $\rho_h^0 = \rho_v^0 = 0.7$ px/cm
(c) $\rho_h^0 = \rho_v^0 = 0.5$ px/cm
(d) $\rho_h^0 = \rho_v^0 = 0.3$ px/cm

FIGURE 6: Visual comparison between fixed Gaussian blur (FGB), AGB [15] and AHGMM on the multi-resolution synthetically generated face data set. (a) Original images with pixel densities decreasing from left to right due different height and pitch angle. (.,.) indicates the horizontal and vertical pixel density in px/cm, respectively. (b-d) For various thresholds (ρ_h^o, ρ_v^o), results of FGB (first row), AGB (second row) and AHGMM filter (third row). For each threshold, FGB is selected w.r.t. the highest pixel density image in the data set. FGB does not adapt its parameters and therefore results into almost blanking out the image with smaller pixel density. In contrast, both AGB and AHGMM maintain high smoothness by varying their parameters depending upon the pixel densities of an image. Comparatively, AGB produces smoother images, while AHGMM filter creates blocking artefacts due to spatial switching of its parameters.

and

$$d = \left\{(h, v) \in \mathbb{Z}^2 : \left\lceil \frac{-\psi_h}{2} \right\rceil \leq h \leq \left\lceil \frac{\psi_h}{2} \right\rceil, \left\lceil \frac{-\psi_v}{2} \right\rceil \leq v \leq \left\lceil \frac{\psi_v}{2} \right\rceil \right\}, \tag{21}$$

with $\psi_j = 2\left\lceil 3\sigma_j \right\rceil + 1$. In order to develop a mixture model from the M discretised Gaussian PSFs of each sub-region, a set of weights ϕ is required. We again utilise a PRNG to generate ϕ such that

$$\phi = \Big\{\phi_{nm} | n \in [1, N], m \in [0, M], \sum_{m=0}^{M} \phi_{nm} = 1\Big\}. \tag{22}$$

Finally, a set of mixture models is generated for each sub-region (line 22 in Algorithm 1) as

$$\mathcal{M} = \Big\{M_n | n \in [1, N]\Big\}, \tag{23}$$

where each element is calculated as

$$M_n = \sum_{m=0}^{M} \phi_{nm} G_{nm}. \tag{24}$$

D. LOCAL AND GLOBAL FILTERING

We have now N discretised Gaussian mixture models in $\mathcal{M}$ for N sub-regions of R. We locally convolve each sub-region R_n (Eq. 12) with their respective M_n to generate a protected sub-region $\bar{R}_n$:

$$\bar{\mathcal{R}} = \left\{ \bar{R}_n | n \in [1, N] \right\}, \tag{25}$$

where $\bar{R}_n = R_n * M_n$. Changing the convolutional kernel for each sub-region generates blocking artefacts (see Fig. 5). To smooth these artefacts, we apply a global convolution filter (line 25 in Algorithm 1) with a Gaussian kernel of zero mean and standard deviation

$$\bar{\sigma}_j = \frac{\sigma_j^o}{Q_j}, \tag{26}$$

where Q_j represents the sub-region size in pixels. As a result, the smoothed protected face $\bar{R}$ is generated to replace R in the original image I thus leading to the protected image I^p. Fig. 6 shows sample images filtered by AHGMM with different thresholds.

E. COMPUTATIONAL COMPLEXITY

The generation of a convolutional kernel is more complex in AHGMM than in the adaptive Gaussian blur filter [15]. In fact, the latter only needs to compute a single Gaussian function, while AHGMM requires the computation of $N \cdot M$ Gaussian functions. Moreover, the adaptive Gaussian blur exploits the separability property of 2D convolutional kernels, i.e. $\psi = \psi_h * \psi_v$, to reduce the number of multiplications and additions from $W \cdot H \cdot |\psi_h| \cdot |\psi_v|$ to $W \cdot H \cdot (|\psi_h| + |\psi_v|)$ (W and H represent the width and height of R in pixels, respectively). Instead, AHGMM dynamically reconfigures the convolutional kernel after processing each sub-region and therefore requires exactly $W \cdot H \cdot |\psi_h| \cdot |\psi_v|$ multiplications and additions.

Algorithm 1 AHGMM

```
     Input: $I$ : unprotected image
            $R$ : detected face region
            $\rho_j$ : pixel density, where $j \in \{h, v\}$
     Output: $I^p$ : protected image
 1:  procedure FilterAHGMM($I, R, \rho_h, \rho_v$)
 2:      for $j = h : v$ do
 3:          $\mu_j^o \leftarrow 0$
 4:          $\sigma_j^o \leftarrow \frac{3\rho_j}{\pi\rho_j^o}$
 5:      end for
 6:      $\mathcal{R} \leftarrow$ $N$ sub-regions of $R$
 7:      for $n = 1 : N$ do
 8:          for $m = 0 : M$ do
 9:              for $j = h : v$ do
10:                  if $m = 0$ then
11:                      $\mu_{jm} \leftarrow \pm\alpha_{jm}\sigma_j^o$
12:                      $\sigma_{jm} \leftarrow (1 \pm \beta_{jm})\sigma_j^o$
13:                  else
14:                      $\mu_{jm} \leftarrow \pm\alpha_{jm}\sigma_j^o\gamma_{jm}$
15:                      $\sigma_{jm} \leftarrow (1 \pm \beta_{jm})\sigma_j^o\gamma_{jm}$
16:                  end if
17:              end for
18:              $X_n \leftarrow (\mu_{jm}, \sigma_{jm})_n$
19:              $G_{nm} \leftarrow$ compute Gaussian PSFs
20:              $\phi_{nm} \leftarrow$ generate weights
21:          end for
22:          $M_n \leftarrow$ Gaussian mixture model
23:          $\bar{R}_n \leftarrow R_n * M_n$
24:      end for
25:      $\bar{R} \leftarrow$ apply global filter on $\bar{\mathcal{R}}$
26:      $I^p \leftarrow$ replace $R$ with $\bar{R}$ in $I$
27:      return $I^p$
28:  end procedure
```

V. DATASET GENERATION

To the best of our knowledge, there exists no large publicly available face dataset collected from an MAV. We therefore generate face images as if they were captured from an MAV via geometric transformation and down-sampling of the LFW dataset [53]. The LFW dataset was collected in an unconstrained environment with extreme illumination conditions and extreme poses. We use the standard verification benchmark test of the LFW dataset (12000 images of 4281 subjects), divided into 10-folds for cross-validation. Each fold contains 600 images of the same subject and 600 images of different subjects. We use the deep funnelled version of the LFW dataset.

Figure 8 shows sample images of the stages of the dataset generation pipeline. We fit a 3D Morphable Model (3DMM) [54] on an input image to detect 68 facial landmarks [55] and then iteratively fit a 3DMM to generate a 3D image representation[2]. As there may be only a few degrees pitch of the subject captured in the images (e.g. a person looking slightly downward or upward), we rotate the 3D image at 0° pitch by applying a geometric transformation computed from the estimated pose of the fitted 3DMM. This

[2] Among the 12000 images, the landmark detector [55] was unable to detect 68 facial landmarks on 74 images. Therefore, we were unable to fit a 3DMM and used the original 74 images in order to comply with the standard verification test script of the LFW data set.

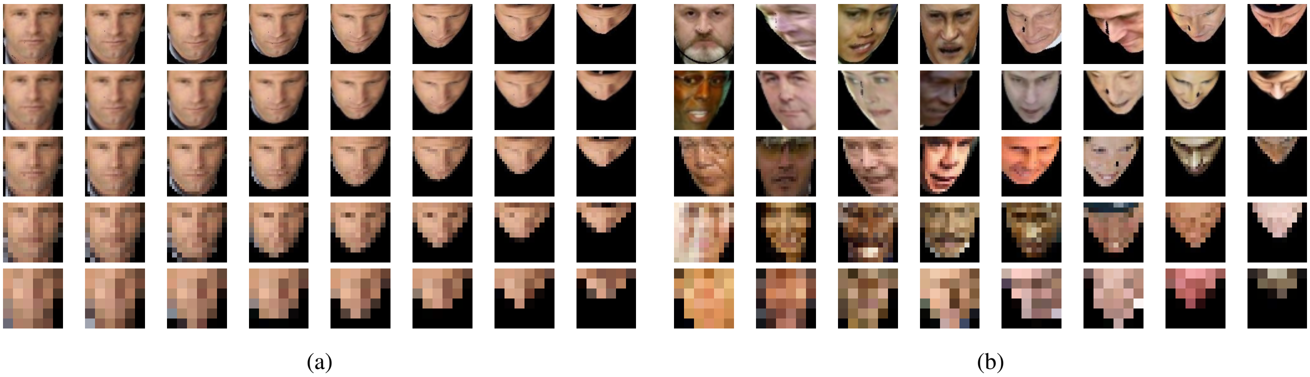


FIGURE 7: Sample images derived from (a) a single subject and (b) different subjects of our synthetically generated airborne data set based on the LFW data set [53]. In each row, the pitch angle varies from 0° to 70° in 10° steps from left to right, while the image resolution remains constant, i.e. first row: 96 × 96 pixels, second row: 48 × 48 pixels, third row: 24 × 24 pixels, fourth row: 12 × 12 pixels and fifth row: 6 × 6 pixels.

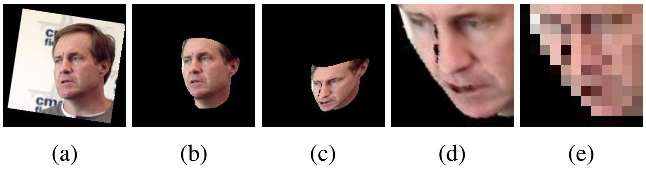


FIGURE 8: Sample images at different stages during the data set generation process. (a) Original image with 250 × 250 pixels, (b) image after fitting a 3D morphable model at 0° pitch angle, (c) image with synthetic pitch effect produced by applying a 3D geometric transformation, (d) aligned image with 96 × 96 pixels produced by applying an affine transformation based on the detected eyes and nose locations and (e) down-sampled image emulating an image captured at a different height.

disturbs the image alignment of the original data set, so a realignment is required, which we perform after generating the pitch effect. The synthetic pitch angles vary from 0° to 70° with a step size of 10° and project it back to generate a corresponding 2D image. In order to align this image so that the eyes and nose appear at the same place among the images belonging to the same pitch angle, we apply an affine transformation computed by detecting eyes and nose tip using the Dlib library [56] such that the transformed face has a resolution of 96 × 96 pixels. As the detection accuracy of the eyes and nose decrease with increasing pitch angle, we generate a ground truth (location of eyes and nose tip) of the 0° pitch angle images and use it for the higher pitch angle images.

Finally, to introduce different height effects for the 8 synthetically generated images, we down-sample them by a factor of 2, 4, 8 and 16 generating images of 48×48, 24×24, 12×12, 6×6 pixels, respectively. Thus, we increase the size of the original standard verification test of the LFW data set by 40 times, i.e. from 12000 images to 480000 images. Fig. 7 shows 40 sample images belonging to the same and different subjects.

We manually determined the values of ρ_h and ρ_v as

$$\rho_h = S_c/S_h, \tag{27}$$

$$\rho_v = S_c cos(\gamma)/S_v, \tag{28}$$

where S_c is the cropped face size in pixels, $\gamma = 90° - \theta_R$ is the pitch angle of the camera, and S_h and S_v are the average human face dimensions, i.e. the bitragion breadth of 15.45 cm and menton-crinion length of 20.75 cm, respectively [57].

VI. EXPERIMENTAL RESULTS

A. EXPERIMENTAL SET UP

We compare AHGMM against Space Variant Gaussian Blur (SVGB) [17], AGB [15] and Fixed Gaussian Blur (FGB), which uses a constant Gaussian kernel defined with respect to the highest resolution face. Thus, we estimate the kernel for FGB as in [15] for the face with 96 × 96 pixels at 0° pitch angle. For the SVGB filter, we divide the face into four concentric circles and reduce the kernel size by 5% while radially moving out between two consecutive regions as in [17]. Although the kernel for the innermost region was manually selected in the original work, we choose the anisotropic kernel as estimated by the AGB [15] and convert it into an isotropic kernel for a fair comparison. We use a block size of 4×4 and $m = 1$ for AHGMM.

To compare privacy filters, we measure the face verfication accuracy using OpenFace [58], an open source implementation of Google's face recognition algorithm FaceNet [59]. OpenFace uses a deep Convolutional Neural Network (CNN) as a feature extractor, which is applied on the training and test images to extract their representations (embeddings) for classification [59]. While the performance of other face recognisers may be slightly better than OpenFace [58] (e.g. SphereFace [60] or ArcFace [61]), their use is not expected to modify the objective of our experiments, which is the analysis of a face recognition algorithm under different privacy attacks.

TABLE 2: Attacks used to evaluate the privacy level of the proposed AHGMM algorithm. Both the gallery faces and the probe faces can be protected or unprotected (naïve-BL). Moreover, the protected faces could be either unchanged or reconstructed (e.g. through an inverse-filter (IF) or super-resolution (SR)). Finally, any AHGMM attack could be further divided into three sub-attacks corresponding to the prior-knowledge of an attacker: optimal, pseudo and accurate.

				Gallery images			
				unprotected	protected		
					unchanged	reconstructed	
						IF	SR
Probe images	unprotected			naïve-BL	N/A	N/A	N/A
	protected	unchanged		naïve-T	parrot-T - optimal - pseudo - accurate	—	—
		reconstructed	IF	naïve-IF -optimal -pseudo -accurate	—	parrot-IF -accurate	—
			SR	naïve-SR -optimal -pseudo -accurate	—	—	parrot-SR -accurate

To measure distortion as in [6, 62], we use the PSNR, the power ratio of the original image with respect to the filtered image.

We perform experiments with 480000 images (consisting of 5 different resolutions and 8 different pitch angles) to determine the validity of the proposed AHGMM to protect the identity information of an individual. For this purpose, we analyse the effect of a naïve-T attack, a parrot-T attack, an inverse filter attack and a super-resolution attack. Moreover, we quantify the corresponding fidelity degradation caused by AHGMM.

Although the synthetic data set include some artifacts (especially for the high pitch angles as, shown in Fig. 7), these do not affect the nature of the analysis, whose objective is to study the relative robustness of privacy filters for a given data set: we use data at the same resolution and pitch angle, including the artifacts, to train and test OpenFace [58]. Moreover, the training set and testing set could be protected or unprotected depending upon the attack type.

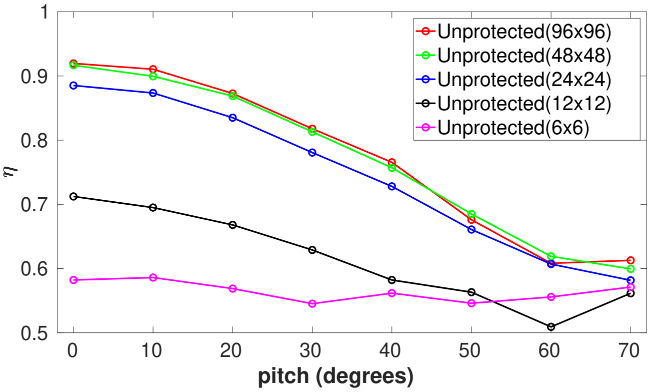


FIGURE 9: Face verification accuracy η of a naïve-BL attack on our synthetically generated face data set. In general, η increases when increasing the face size except at high pitch angles of 60 and 70 degrees where it slightly fluctuates. For 6×6 pixels faces, η is the lowest and rather independent of the pitch angle.

As AGB and SVGB do not use any secret key, we evaluate them only using their accurate parameters in the parrot-T, inverse filter and super-resolution attacks. In contrast, any of these attacks on AHGMM can be further divided into three sub-attacks: optimal kernel, pseudo AHGMM and accurate AHGMM. In the optimal kernel sub-attack, we assume that an attacker is able to estimate the parameters of the optimal kernel and applies the optimal kernel to the entire face. In the pseudo AHGMM sub-attack, we assume that the attacker knows the optimal kernel and randomly modifies the filter parameter for the N sub-regions. In the accurate AHGMM sub-attack, we assume that the attacker has access to the secret key and can decipher all the parameters of the filter for the N sub-regions. As this prior knowledge can be exploited for both probe and gallery images, we evaluate AHGMM under 13 different scenarios (see Table 2).

We assume that an attacker is able to determine the pitch angle of a protected face using the background information of an image captured from an MAV and can apply a geometric transformation to modify the gallery images at that pitch angle. Therefore, in all the following attacks, both the gallery and the probe images are at the same pitch angle that can be protected or unprotected depending upon the attack type. Moreover, we use the same resolution for both the gallery images and the probe images.

B. NAÏVE-T ATTACK

First of all, we perform a naïve-BL attack which shows the baseline face verfication accuracy when both the probe data set and the gallery data set are unprotected. The results of the naïve-BL attack are given in Fig. 9. Then we perform a naïve-T attack in which the gallery images are unprotected, while the probe images are protected using FGB, SVGB [17], AGB [15] and AHGMM. The results of this attack are given in Fig. 10 at different thresholds ρ_j^o.

The naïve-BL attack shows that the accuracy η of our synthetically generated data set decreases with the decrease of the face resolution and with the increase in the face pitch angle. However, this trend vanishes at high pitch angles, i.e. 60° and 70°, where it shows a slight randomness. Finally, for the low resolution faces (6×6 pixels), the accuracy does not show any effect of the pitch angle and slightly oscillates. Therefore, we consider 6×6 pixels inherently privacy protected and remove these images from the analysis of the privacy filters.

From the naïve-T attack, we are interested in finding the optimal threshold which defines the optimal kernel for AGB [15] (see Section IV and Eq. 11). It is clear from

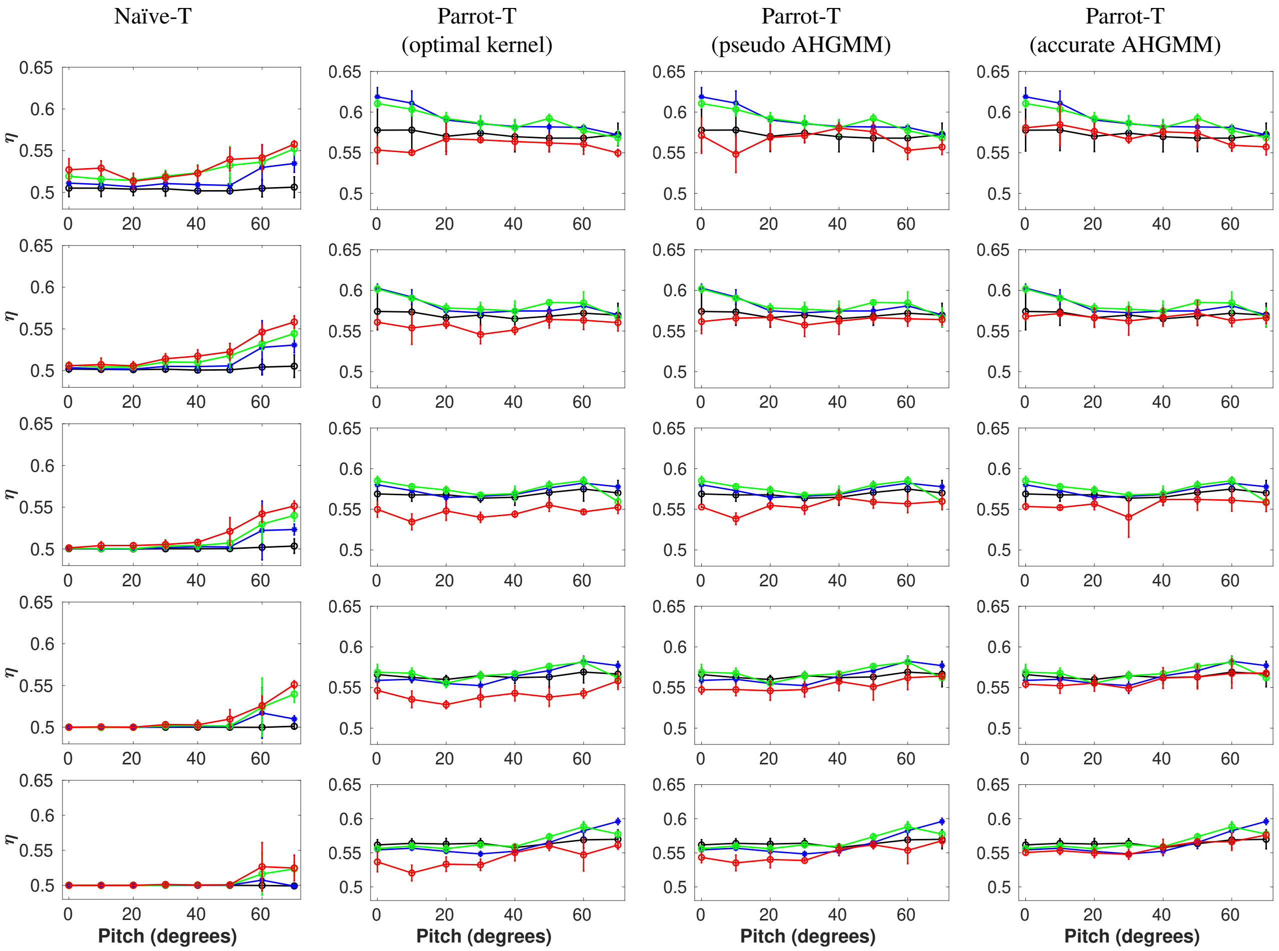


FIGURE 10: Face verification accuracy η achieved by naïve and parrot attacks on images protected by four different privacy protection filters at different thresholds ρ_j^o: first row: $\rho_j^o = 0.7$ px/cm, second row: $\rho_j^o = 0.6$ px/cm, third row: $\rho_j^o = 0.5$ px/cm, fourth row: $\rho_j^o = 0.4$ px/cm, fifth row: $\rho_j^o = 0.3$ px/cm. The filled marker shows the mean and the vertical bar indicates the standard deviation of η for the multi-resolution images (96×96, 48×48, 24×24, 12×12). Legend: 🟥 AHGMM, 🟩 AGB [15], 🟦 SVGB [17], ⬛ FGB. Under the naïve-T attack, AHGMM posses the highest η which converges towards $\eta = 0.5$ as the ρ_j^o is decreased and finally at $\rho_j^o \leq 0.5$ px/cm, the difference between η of AHGMM, AGB, SVGB and FGB becomes negligible, except unexpectedly at pitch angles 60° and 70° degrees. The parrot-T attack on AHGMM is divided into three sub-attacks: optimal kernel parrot-T attack, pseudo AHGMM parrot-T attack and accurate AHGMM parrot-T attack. In contrast to naïve-T attack, AHGMM provides the lowest η under any type of the three parrot-T attacks and this fact becomes negligible at $\rho_j^o = 0.3$ px/cm under accurate AHGMM parrot-T attack.

Fig. 10 that the accuracy of the naïve-T attack decreases while decreasing the threshold. When the threshold reaches 0.5 px/cm, the difference between the accuracy achieved by AGB [15] and a random classifier ($\eta = 0.5$) becomes very small except, unexpectedly, at high pitch angles. This difference further decreases at 0.4 px/cm and 0.3 px/cm. Thus, the optimal threshold defining the optimal kernel can be 0.5 px/cm, 0.4 px/cm and 0.3 px/cm. The last two thresholds decrease the accuracy negligibly, but distort the images severely. Therefore, we analysed the trade-off of accuracy (under naïve, parrot attack and reconstruction attacks) and distortion for these three thresholds.

At these three thresholds under the naïve-T attack, the accuracy of AHGMM is higher as compared to the AGB [15]. The main reason for this slightly higher accuracy is due to the under blurred sub-regions of the AHGMM filtered face as it hops its kernel below and above the optimal Gaussian kernel. In contrast, the accuracy of the Space Variant Gaussian Blur [17] is always lower than AGB and AHGMM. This is because SVGB uses an isotropic Gaussian kernel which deteriorates a face more severely as compared to the anisotropic kernel of the AGB and AHGMM filter. FGB has the lowest accuracy at any threshold due to over blurring of all images except 96×96 pixels images at 0° pitch angle.

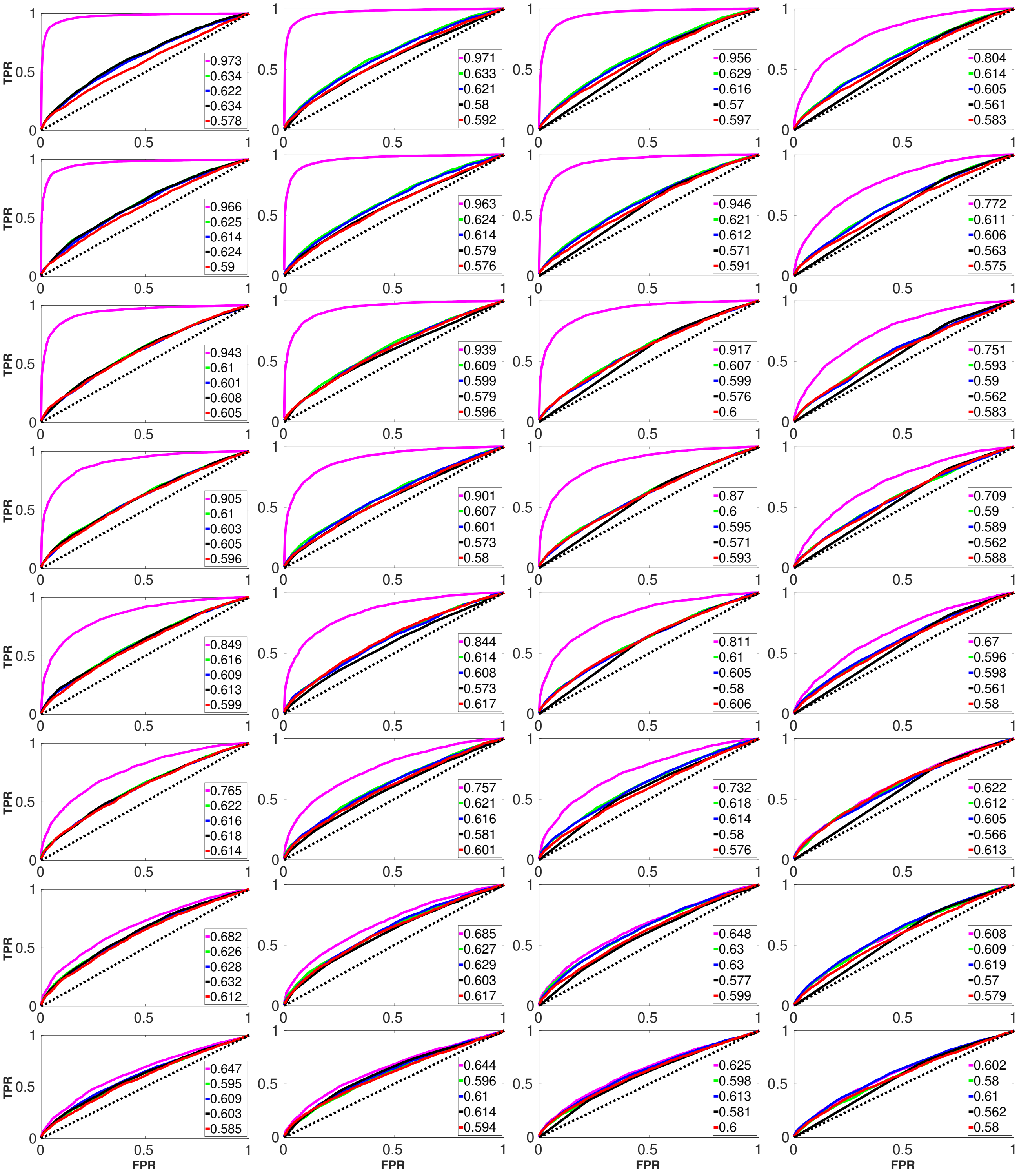


FIGURE 11: Receiver Operating Curves (ROCs) for the accurate AHGMM parrot-T attack at threshold $\rho_j^o = 0.5$ px/cm. Each ROC is the mean of 10-curves generated by the 10-folds used for cross validation. Legend: — Unprotected, — AGB, — SVGB, — FGB, — AHGMM. In each column, the image resolution remains constant, i.e. first column: 96×96, second column: 48×48, third column: 24×24 and fourth column: 12×12 pixels, while the pitch angle varies i.e. first row: $0°$, second row: $10°$, third row: $20°$, fourth row: $30°$, fifth row: $40°$, sixth row: $50°$, seventh row: $60°$ and eighth row: $70°$. The legend values represent the Area Under Curve (AUC).

C. PARROT-T ATTACK

In the parrot-T attack, we filter both gallery and probe images and then evaluate the achieved accuracy. We study the parrot-T attack on AHGMM under three sub-attacks: optimal kernel parrot-T sub-attack, pseudo-AHGMM parrot-T sub-attack and accurate AHGMM parrot-T sub-attack. The accuracy results of these sub-attacks are given in Fig. 10 at different thresholds ρ_j^o, while Receiver Operating Curves (ROCs) for the accurate AHGMM parrot-T sub-attack at $\rho_j^o = 0.5$ px/cm are presented in Fig. 11.

The parrot-T attack on state-of-the-art privacy filters increases the accuracy as compared to the naïve-T attack. Under the optimal kernel parrot sub-attack, our AHGMM shows the least accuracy improvement at any of the three thresholds. This is because the optimal kernel Gaussian blur is a spatially invariant blur that is not helpful in recognising spatially varying Gaussian blurred images, e.g. the AHGMM filtered images. Thus, our AHGMM provides the lowest accuracy against the parrot-T attack using the optimal kernel.

The pseudo AHGMM parrot-T sub-attack slightly improves the accuracy further as compared to the optimal kernel parrot-T sub-attack. The main reason is that both the gallery and the probe images are now filtered using spatially varying Gaussian blur. However, under the pseudo AHGMM sub-attack, the accuracy of AHGMM remains below the other three state-of-the-art privacy filters. Thus, our AHGMM provides the highest privacy protection even against the pseudo AHGMM parrot sub-attack.

Finally, the accurate AHGMM sub-attack improves the accuracy as compared to the optimal kernel and almost eqivalent to the pseudo AHGMM sub-attacks. Comparatively, even under the accurate AHGMM sub-attack, AHGMM performs better than FGB, AGB [15] and SVGB [17] at these three thresholds with the least improvement at $\rho_j^o = 0.3$ px/cm.

From the accurate AHGMM sub-attack, it is apparent that AHGMM permanently removes the sensitive information from the face and an attacker can not recognise it with a high accuracy even when he/she has access to the secret key. This is in contrast to the reversible filters, e.g. encryption/scrambling based filters, which can reconstruct the original face after having the secret key. Thus, AHGMM is robust against a brute-force attack.

D. INVERSE FILTER ATTACK

In the inverse-filter (IF) attack, we reconstruct the probe images by deconvolving the protected face with an accurate or estimated kernel. We evaluate the IF attack under four sub-attacks: optimal kernel naïve-IF sub-attack, pseudo AHGMM naïve-IF sub-attack, accurate AHGMM naïve-IF sub-attack and accurate AHGMM parrot-IF sub-attack. Fig. 12 depicts the effect of inverse filtering on selected sample images protected with AGB, SVGB and AHGMM. Fig. 13 shows the achieved accuracies under the different sub-attacks at different values of ρ_j^o, while Fig. 14 presents ROCs for the accurate AHGMM parrot-IF sub-attack at $\rho_j^o = 0.5$ px/cm.

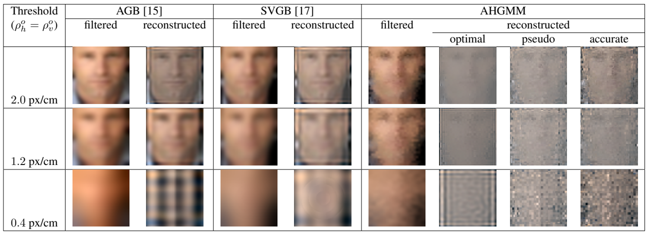


FIGURE 12: Inverse filtering of protected faces at different thresholds ρ_j^o. AGB and SVGB protected faces can be reconstructed by inverse filtering to some extent. Inverse filtering of AHGMM protected faces is hardly possible even if the hopping kernel parameters are known.

TABLE 3: Face verification accuracy η after a super-resolution attack on faces protected by adaptive Gaussian blur (AGB), space variant Gaussian blur (SVGB) and AHGMM at threshold $\rho_j^o = 0.5$ px/cm. The values of η are given as $\tilde{\mu}(\tilde{\sigma})$, where $\tilde{\mu}$ indicates the mean and $\tilde{\sigma}$ the standard deviation for the 10-fold cross validations. In the naïve-SR attack, the reconstructed probe faces are compared against the unprotected gallery images, while both the probe and the gallery images are super-resolved in the parrot-SR attack.

Attack type	AGB	SVGB	AHGMM
optimal naïve-SR	0.592 (0.012)	0.566 (0.016)	**0.515 (0.014)**
pseudo AHGMM naïve-SR	–	–	**0.520 (0.006)**
accurate AHGMM naïve-SR	–	–	**0.532 (0.018)**
accurate AHGMM parrot-SR	0.634 (0.015)	0.583(0.034)	**0.546 (0.018)**

As can be seen in Fig. 12, the face reconstruction quality decreases when the threshold increases (increasing the filter kernel) even if the filter parameters are known. This is true for both space invariant Gaussian blur (i.e. AGB) and linear space variant Gaussian blur (i.e. SVGB). The main reason is that the boundaries of the face start propagating towards the centre of the face as the threshold is decreased. Thus, it becomes difficult to distinguish between reconstructed faces at the lower thresholds (see Fig. 13).

In case of non-linear space variant blur (AHGMM), the reconstruction becomes more challenging even when the same hopping kernels are used as for the protection. The main reason, in addition to the boundary propagation, is that while deconvolving a sub-region, the IF incorrectly treats the adjacent subregions as if they were filtered with the same kernel, thus not enabling it to reconstruct the original face (see Fig. 12). Consequently, it becomes difficult to accurately predict the label of the reconstructed face.

In contrast to naïve-IF attacks, parrot-IF attack is more severe and increases significantly the accuracy, especially for AGB, FGB and SVGB. AHGMM also shows the accuracy improvement but less than AGB, FGB and SVGB; and is more robust to an inverse filter attack even when using an accurate secret key.

E. SUPER-RESOLUTION ATTACK

In this attack, we reconstruct the filtered probe images with SRCNN [8]. SRCNN first learns a mapping between

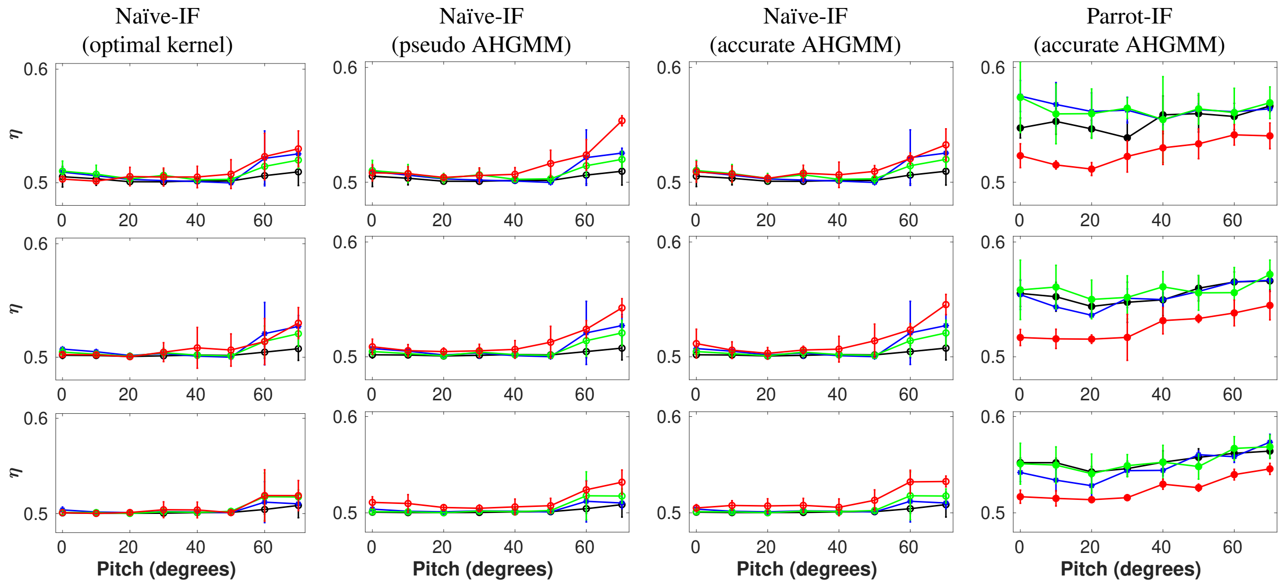


FIGURE 13: Face verification accuracy η achieved by an inverse filter (IF) attack on images protected by four different privacy protection filters at different thresholds ρ_j^o: first row: $\rho_j^o = 0.7$ px/cm, second row: $\rho_j^o = 0.6$ px/cm, third row: $\rho_j^o = 0.5$ px/cm. The filled marker shows the mean and the vertical bar indicates the standard deviation of η for the multi-resolution images (96×96, 48×48, 24×24, 12×12). Legend: — AHGMM, — AGB, — SVGB, — FGB. The IF attack is investigated under four sub-attacks: optimal kernel naïve-IF, pseudo AHGMM naïve-IF, accurate AHGMM naïve-IF and accurate AHGMM parrot-IF attack. AHGMM achieves a slightly higher η under the naïve-IF attacks than the state-of-the-art filters, independently of the used threshold ρ_j^o. In contrast, AHGMM achieves the lowest η under the parrot-IF attack. As η is close to 0.5 under the naïve-IF attack for $0.5 \leq \rho_j^o \leq 0.7$ px/cm, we therefore do not perform experiments for $\rho_j^o < 0.5$ px/cm.

the high-resolution images and their corresponding low-resolution version, and then applies this mapping to enhance the details of a low-resolution image. We learn the SRCNN mapping for 1,000,000 iterations between the protected images (i.e. the low resolution) and their corresponding unprotected images (i.e. the high resolution) using the same data sets (91-images and Set5) as used in [8]. Such a mapping is separately learned for each privacy filter (i.e. AGB, SVGB, and AHGMM). As learning the mapping is a time consuming process, we investigate the super-resolution attack for a single point of our synthetic data set: 12000 images each with 96×96 pixels and 0° pitch angle.

We evaluate the super-resolution (SR) attack under four sub-attacks: optimal kernel naïve-SR sub-attack, pseudo AHGMM naïve-SR sub-attack, accurate AHGMM naïve-SR sub-attack and accurate AHGMM parrot-SR sub-attack. Tab. 3 summarises the accuracies under the different sub-attacks, while Fig. 15 depicts a visual comparison of the super-resolution reconstruction for three sample faces protected by AGB, SVGB and AHGMM filters. Fig. 16 presents the ROC for the accurate AHGMM parrot-SR sub-attack.

For the space-invariant Gaussian blur (AGB), it is apparent from Fig. 15 that the SR attack can reconstruct the faces more effectively, even when the kernel size is quite high (i.e. $\rho_j^o = 0.5$ px/cm). Therefore, the faces protected by AGB achieves a higher accuracy (see Tab. 3). In contrast, faces protected by linear space variant Gaussian blur (SVGB) are difficult to reconstruct. The main reason is that the SR mapping becomes erroneous especially for patches which contain parts processed by different kernels. However, SR can effectively reconstruct patches where the Gaussian blur is locally invariant (e.g. compare the areas around the eyes of the SVGB restored faces in Fig. 15). The overall reconstruction is worse than for AGB and thus the achieved accuracy is lower.

Reconstruction by super-resolution is even more challenging for AHGMM protected faces. The main reason is that a single patch for learning the mapping contains several sub-regions each filtered with pseudo-randomly correlated Gaussian mixture models. Thus, the error in the learned SR mapping increases thus resulting in the lowest accuracy as compared to AGB and SVGB.

Similarly to the parrot-IF attack, the accuracy improves for the parrot-SR attack where SR-reconstruction is also performed for the gallery images. Especially for AGB and SVGB, the similarity between (protected and reconstructed) gallery images and the (reconstructed) probe images increases. Thus, the accuracy increases. As for the other attacks, AHGMM is more robust to parrot attacks than AGB and SVGB, and achieves the lowest accuracy.

F. DISTORTION ANALYSIS

We measure the distortion of FGB, SVGB [17], AGB [15] and AHGMM using PSNR. For a trade-off analysis between

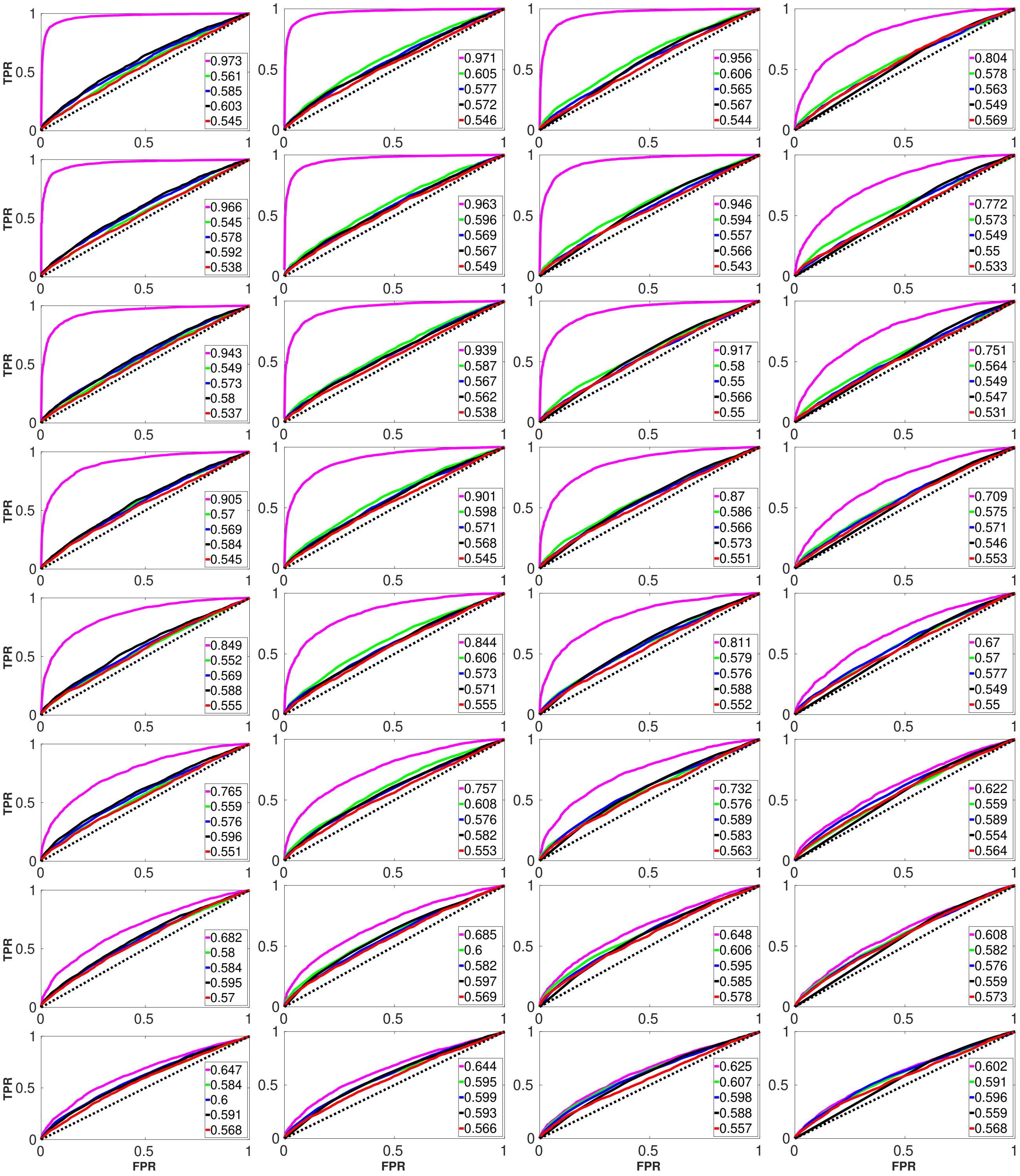


FIGURE 14: Receiver Operating Curves (ROCs) for the accurate AHGMM parrot-IF attack at threshold $\rho_j^o = 0.5$ px/cm. Each ROC is the mean of 10-curves generated by the 10-folds used for cross validation. Legend: — Unprotected, — AGB, — SVGB, — FGB, — AHGMM. In each column, the image resolution remains constant, i.e. first column: 96 × 96, second column: 48 × 48, third column: 24 × 24 and fourth column: 12 × 12 pixels, while the pitch angle varies i.e. first row: 0°, second row: 10°, third row: 20°, fourth row: 30°, fifth row: 40°, sixth row: 50°, seventh row: 60° and eighth row: 70°. The legend values represent the Area Under Curve (AUC).

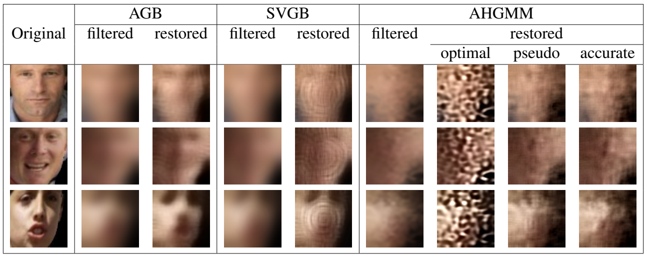


FIGURE 15: Visual comparison of reconstructed faces with super-resolution algorithm SRCNN [8] for threshold $\rho_j^o = 0.5$ px/cm. The reconstruction performance deteriorates from AGB [15] over SVGB [17] to AHGMM protected faces.

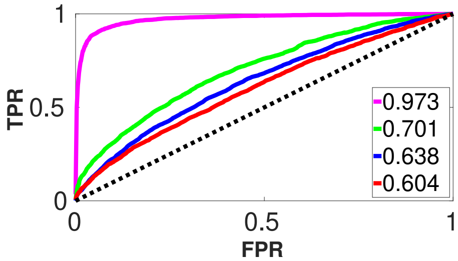


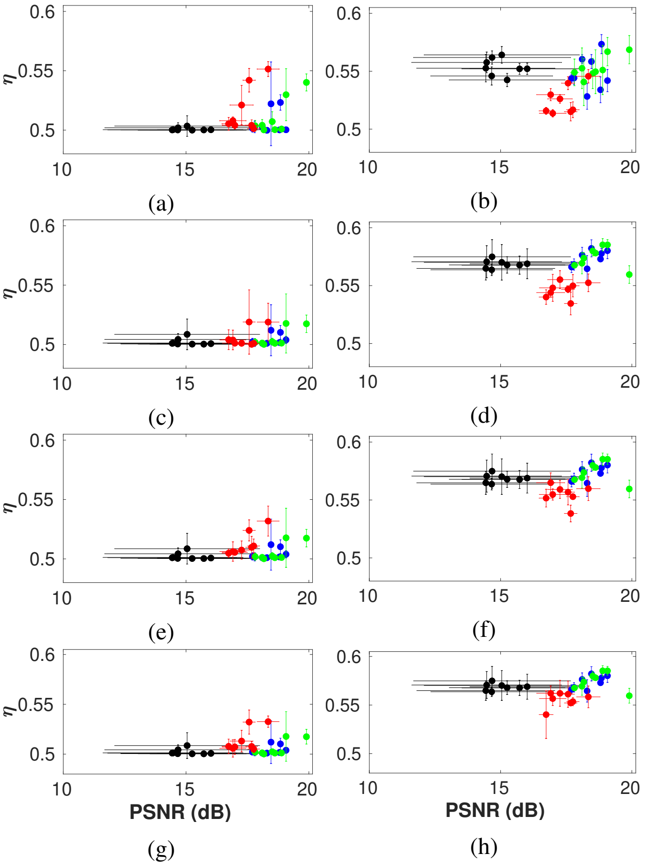
FIGURE 16: Receiver Operating Curve (ROC) for the accurate AHGMM parrot-SR attack at threshold $\rho_j^o = 0.5$ px/cm. Each ROC is the mean of 10-curves generated by the 10-folds used for cross validation. Legend: — Unprotected, — AGB, — SVGB, — AHGMM. This test is performed only for a single resolution (96 × 96 pixels) and pitch angle (0°). The legend values represent the Area Under Curve (AUC).

distortion and privacy, we plot the face verification accuracy against PSNR. The results of this trade-off analysis are presented in Fig. 17.

AGB [15] has the highest average PSNR values followed by SVGB [17], AHGMM and FGB. The main reason is that AGB uses a single anisotropic kernel instead of the spatially and linearly varying kernel used by SVGB [17]. Although AHGMM also uses an anisotropic kernel like AGB, the spatial hopping phenomena of the Gaussian mixture model of AHGMM result in high distortion (PSNR values) as compared to AGB and SVGB (see Fig. 6). FGB has the highest distortion as it does not change its parameters depending upon the resolution of the face.

VII. CONCLUSION

We presented an irreversible visual privacy protection filter that is robust against parrot, inverse-filter and super-resolution attacks on protected image regions. The proposed filter uses an adaptive hopping Gaussian mixture model and adapts its parameters depending upon the resolution of sensitive regions in order to minimise distortion, while locally and pseudo-randomly hopping them to prevent an attacker from estimating them. We evaluated AHGMM using a state-of-the-art face recognition algorithm and a synthetic face data set with faces at different pitch angles and resolutions that emulate faces as captured from an MAV. The proposed algorithm provides the highest privacy protection under a parrot, an inverse-filter and a super-resolution attack and an almost equivalent level of privacy to state-of-the-art privacy filters under a naïve attack. Unlike face-de-identification



(a) (b) (c) (d) (e) (f) (g) (h)

FIGURE 17: Trade-off analysis between the face verification accuracy η and the distortion provided by the different privacy filters under the naïve-T, parrot-T and inverse filter (IF) attacks at threshold $\rho_j^o = 0.5$ px/cm. The distortion is measured by the Peak Signal to Noise Ratio (PSNR). Legend: — AHGMM, — AGB [15], — SVGB [17], — FGB. Under the naïve-T attack, our proposed AHGMM possesses η almost equivalent to the state-of-the-art filter, but lowest under the parrot-T attacks. However, AHGMM has slightly lower PSNR as compared to AGB and SVGB, but much higher than FGB. (a) naïve-T attack, (b) accurate AHGMM parrot-IF attack, (c) optimal kernel naïve-IF attack, (d) optimal kernel parrot-T attack, (e) pseudo AHGMM naïve-IF attack, (f) pseudo AHGMM parrot-T attack, (g) accurate AHGMM naïve-IF attack and (h) accurate AHGMM parrot-T attack. For the last three naïve-IF and parrot-T attacks, the results of AGB, SVGB and FGB are the same and have been superimposed for the comparison. Please see Section VI-B, Section VI-C and Section VI-D for the details of the attacks.

TABLE 4: Notation.

Notation	Meaning
$R, \bar{R}, \hat{R}$	*unprotected, protected and reconstructed face region*
C_R	*centre of* R
W, H	*width and height of* R
$\mathcal{D}$	*data set including both gallery and probe data sets*
$\mathcal{R}_{\mathcal{G}}, \bar{\mathcal{R}}_{\mathcal{G}}, \hat{\mathcal{R}}_{\mathcal{G}}$	*unprotected, protected and reconstructed gallery data set*
$\mathcal{R}_{\mathcal{P}}, \bar{\mathcal{R}}_{\mathcal{P}}, \hat{\mathcal{R}}_{\mathcal{P}}$	*unprotected, protected and reconstructed probe data set*
$K, \tilde{K}$	*original and predicted identity labels*
F_{Ω_j}	*privacy filter of parameter* Ω_j
G	*function that an attacker exploits*
D	*distortion introduced by* F_{Ω_j}
P	*probability of predicting the label of a face*
η	*face verification accuracy*
ϵ	*verification accuracy of a random classifier*
f	*focal length of the camera*
p_j	*physical dimension of a pixel in* j *direction*
h_1, h_2	*height of a camera and face from ground level*
$\boldsymbol{N}, \boldsymbol{P}$	*vectors representing Nadir and principal axis of a camera*
θ_R, θ_P	*angle between* $\boldsymbol{N}$, R *and* $\boldsymbol{N}$, $\boldsymbol{P}$
N	*number of sub-regions of* R
M	*number of supplementary Gaussian functions*
ρ_j	*pixel density (px/cm)*, where $j \in \{h, v\}$
ρ_j^o	*threshold pixel density for privacy filtering*
μ_j, σ_j	*mean and standard deviation of a Gaussian PSF*
μ_j^o, σ_j^o	*mean and standard deviation of an optimal Gaussian PSF*
μ_{jm}, σ_{jm}	*randomly modified* μ_j^o *and* σ_j^o *for* m^{th} *Gaussian PSF*
α_{jm}, β_{jm}	*randomly generated numbers for* μ_{jm} *and* σ_{jm}
$\Omega_j, \Omega_j^o, \bar{\Omega}_j^o$	*tuple* (μ_j, σ_j), (μ_j^o, σ_j^o) and (μ_{jm}, σ_{jm})
f_s, f_s^o	*Nyquist frequency of* ρ_j *and* ρ_j^o
$\acute{\sigma}_j^o$	*frequency domain standard deviation corresponding to* σ_j^o
γ_{jm}	*scaling factor for* σ_j^o
$\mathcal{X}$	*set of tuple containing the parameters of Gaussian functions*
$\mathcal{G}$	*set of Gaussian functions*
G_{nm}	*element of* $\mathcal{G}$
ϕ	*set of weights for a Gaussian mixture model*
ϕ_{nm}	*element of* ϕ
$\mathcal{M}$	*Gaussian mixture model*
M_n	*element of* $\mathcal{M}$
Q_j	*sub-region size in pixels*
$\bar{\sigma}_j$	*standard deviation of a global smoothing filter*

approaches ([4, 16, 35–37, 42–45, 63]), we do not depend on an auxiliary visual detector (i.e. pose, facial expression, age, gender, race) to counter a parrot, an inverse-filter or a super-resolution attack. Moreover, unlike encryption/scrambling filters ([3, 5, 10–12, 23–27, 29, 30]), we prevent the recovery of the original values of the protected region even with access to the seed of the PRNG.

A work that extends the proposed privacy filter to videos and that removes the jitter introduced by the proposed filter is presented in [64].

ACKNOWLEDGMENT

O. Sarwar was supported in part by Erasmus Mundus Joint Doctorate in Interactive and Cognitive Environment, which is funded by the Education, Audio-visual & Culture Executive Agency under the FPA no 2010-0015. This work was also supported by the research initiative Mobile Vision Austria with funding from the Austrian Federal Ministry of Science, Research and Economy, and the Austrian Institute of Technology. Andrea Cavallaro wishes to thank the Alan Turing Institute (EP/N510129/1), which is funded by the EPSRC, for its support through the project PRIMULA.

REFERENCES

[1] J. Schiff, M. Meingast, D. K. Mulligan, S. Sastry, and K. Goldberg, “Respectful cameras: detecting visual markers in real-time to address privacy concerns,” in Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, San Diego, USA, Oct. 2007, pp. 971–978.

[2] D. Kundur and D. Hatzinakos, “Blind image deconvolution,” IEEE Signal Processing Magazine, vol. 13, no. 3, pp. 43–64, May 1996.

[3] T. E. Boult, “PICO: Privacy through invertible cryptographic obscuration,” in Proc. Computer Vision for Interactive and Intelligent Environment, Lexington, USA, Nov. 2005, pp. 27–38.

[4] E. M. Newton, S. L. Sweeney, and S. B. Malin, “Preserving privacy by de-identifying facial images,” IEEE Trans. on Knowledge and Data Engineering, vol. 17, pp. 232–243, Feb. 2005.

[5] F. Dufaux and T. Ebrahimi, “Scrambling for privacy protection in video surveillance systems,” IEEE Trans. on Circuits and Systems for Video Technology, vol. 18, no. 8, pp. 1168–1174, Aug. 2008.

[6] A. Erdelyi, T. Barat, P. Valet, T. Winkler, and B. Rinner, “Adaptive cartooning for privacy protection in camera networks,” in Proc. Int. Conf. on Advanced Video and Signal-based Surveillance, Seoul, Korea, Aug. 2014, pp. 44–49.

[7] P. Korshunov and T. Ebrahimi, “Towards optimal distortion-based visual privacy filters,” in Proc. IEEE Int. Conf. on Image Processing, Paris, France, Oct. 2014, pp. 6051–6055.

[8] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[9] S. R. Michael, R. Brandon, F. Charles, and J. Y. Hyun, “Privacy-preserving human activity recognition from extreme low resolution,” in Proc. AAAI Conf. on Artificial Intelligence, San Francisco, California, USA, Feb. 2017, pp. 1–8.

[10] X. Zhang, S. Seo, and C. Wang, “A lightweight encryption method for privacy protection in surveillance videos,” IEEE Access, vol. 6, pp. 18 074–18 087, Apr. 2018.

[11] C. Ma, C. Yan, and C. W. Chen, “Scalable access control for privacy-aware media sharing,” IEEE Trans. on Multimedia, vol. 21, no. 1, pp. 173–183, Jan. 2019.

[12] S. Ullah, L. Marcenaro, and B. Rinner, “Secure smart cameras by aggregate-signcryption with decryption fairness for multi-receiver IoT applications,” Sensors, vol. 19, no. 2, pp. 1–22, Jan. 2019.

[13] Y. Kim, J. Jo, and S. Shrestha, “A server-based real-time privacy protection scheme against video surveillance by unmanned aerial systems,” in Proc. Int. Conf. on Unmanned Aircraft Systems, Orlando, USA, May 2014, pp. 684–691.

[14] R. Babiceanu, P. Bojda, R. Seker, and M. Alghumgham, “An onboard UAS visual privacy guard system,” in Proc. Integrated Communication, Navigation, and Surveillance Conf., Herdon, USA, Apr. 2015, pp. 1–8.

[15] O. Sarwar, B. Rinner, and A. Cavallaro, “Design space exploration for adaptive privacy protection in airborne images,” in Proc. IEEE Advanced Video and Signal-based Surveillance, Colorado Springs, USA, Aug. 2016, pp. 159–165.

[16] L. Du, M. Yi, E. Blasch, and H. Ling, “Garp-face: Balancing privacy protection and utility preservation in face de-identification,” in Proc. IEEE Int. Joint Conf. on Biometrics, Clearwater, Florida,USA, Sep. 2014, pp. 1–8.

[17] M. Saini, P. K. Atrey, S. Mehrotra, and M. Kankanhalli, “Adaptive transformation for robust privacy protection in video surveillance,” Advances in Multimedia, vol. 2012, pp. 1–14, Feb. 2012.

[18] Safe Haven, Safe Haven from Iceberg Systems ensures privacy from camera phones; Camera phone voyeurs and spy‘s can be defeated by new technology, http://www.m2.com/m2/web/story.php/20031E97F470D8BB9F6685256D9D006B597D, 2003, [Last accessed: 2019-07-21].

[19] Eagle Eye, Bulletin of the Connecticut Academy of Science and Engineering., vol. 12, no. 2, 1997.

[20] S. Zhu, C. Zhang, and X. Zhang, “Automating visual privacy protection using a smart LED,” in Proc. Int. Conf. on Mobile Computing and Networking, Snowbird, Utah, USA, Oct. 2017, pp. 329–342.

[21] Y. Zhang, Y. Lu, H. Nagahara, and R.-i. Taniguchi, “Anonymous camera for privacy protection,” in Proc. Int. Conf. on Pattern Recognition, Stockholm, Sweden, Aug. 2014, pp. 4170–4175.

[22] Z. W. Wang, V. Vineet, F. Pittaluga, S. N. Sinha, O. Cossairt, and S. Bing Kang, “Privacy-preserving action recognition using coded aperture videos,” in Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, June 2019, pp. 1–10.

[23] A. Chattopadhyay and T. E. Boult, “PrivacyCam: A privacy preserving camera using uCLinux on the blackfin DSP,” in Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Minneapolis, USA, June 2007, pp. 1–8.

[24] S. Rahman, M. Hossain, H. Mouftah, A. El Saddik, and E. Okamoto, “A real-time privacy-sensitive data hiding approach based on chaos cryptography,” in Proc. IEEE Int. Conf. on Multimedia and Expo, Suntec City, Singapore, July 2010, pp. 72–77.

[25] T. Winkler and B. Rinner, “Securing embedded smart cameras with trusted computing,” EURASIP Journal on Wireless Communications and Networking, vol. 2011, pp. 1–20, Jan. 2011.

[26] N. Baaziz, N. Lolo, O. Padilla, and F. Petngang, “Security and privacy protection for automated video surveillance,” in Proc. IEEE Int. Symposium on Signal Processing and Information Technology, Cairo, Egypt, Dec. 2007, pp. 17–22.

[27] H. Sohn, D. N. Wesley, and Y. Man Ro, “Privacy protection in video surveillance systems: Analysis of subband-adaptive scrambling in JPEG XR,” IEEE Trans. on Circuits and Systems for Video Technology, vol. 21, no. 2, pp. 170–177, Feb. 2011.

[28] N. Ruchaud and J. L. Dugelay, “ASePPI: Robust privacy protection against de-anonymization attacks,” in Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops, Honolulu, Hawaii, US, July 2017, pp. 1352–1359.

[29] P. Korshunov and T. Ebrahimi, “Using warping for privacy protection in video surveillance,” in Proc. Int. Conf. on Digital Signal Processing, Fira, Santorini, Greece, July 2013, pp. 1–6.

[30] ——, “Using face morphing to protect privacy,” in Proc. IEEE Int. Conf. on Advanced Video and Signal-based Surveillance, Kraków, Poland, Aug. 2013, pp. 208–213.

[31] R. Jiang, S. Al-Maadeed, A. Bouridane, D. Crookes, and M. Celebi, “Face recognition in the scrambled domain via salience-aware ensembles of many kernels,” IEEE Trans. on Information Forensics and Security, vol. 11, no. 8, pp. 1807–1817, Aug. 2016.

[32] R. Jiang, A. Bouridane, D. Crookes, M. Celebi, and H. L. Wei, “Privacy-protected facial biometric verification using fuzzy forest learning,” IEEE Trans. on Fuzzy Systems, vol. 24, no. 4, pp. 779–790, Aug. 2016.

[33] H. Rashwan, M. García, A. Ballesté, and D. Puig, “Defeating face de-identification methods based on DCT-block scrambling,” Machine Vision and Applications, vol. 27, pp. 251–262, Dec. 2015.

[34] M. Koelle, S. Ananthanarayan, S. Czupalla, W. Heuten, and S. Boll, “Your smart glasses’ camera bothers me!: Exploring opt-in and opt-out gestures for privacy mediation,” in Proc. Nordic Conf. on Human-Computer Interaction, Oslo, Norway, Oct. 2018, pp. 473–481.

[35] R. Gross, S. L. Sweeney, F. Torre, and S. M. Baker, “Model-based face de-identification,” in Proc. Conf. on Computer Vision and Pattern Recognition Workshop, New York, USA, June 2006, pp. 161–168.

[36] Y. Lin, S. Wang, Q. Lin, and F. Tang, “Face swapping under large pose variations: A 3D model based approach,” in Proc. IEEE Int. Conf. on Multimedia and Expo, Melbourne, Australia, July 2012, pp. 333–338.

[37] G. Letournel, A. Bugeau, V. T. Ta, and J. P. Domenger, “Face de-identification with expressions preservation,” in Proc. IEEE Int. Conf. on Image Processing, Quebec, Canada, Sep. 2015, pp. 4366–4370.

[38] A. Jourabloo, X. Yin, and X. Liu, “Attribute preserved face de-identification,” in Proc. Int. Conf. on Biometrics, Phuket, Thailand, May 2015, pp. 278–285.

[39] R. Zhongzheng, J. L. Yong, and S. R. Micheal, “Learning to anonymize faces for privacy preserving action detection,” in Proc. European Conf. on Computer Vision, Munich, Germany, Sep. 2018, pp. 1–17.

[40] B. Meden, Z. Emersic, V. Struc, and P. Peer, “k-Same-Net: k-Anonymity with generative deep neural networks for face de-identification,” Entropy, vol. 20,

no. 1, pp. 1–24, Jan. 2018.

[41] W. Yifan, Y. Fan, X. Yong, and L. Haibin, "Privacy-protective-GAN for privacy preserving face de-identification," Journal of Computer Science and Technology, vol. 34, no. 1, pp. 1–13, Jan. 2019.

[42] J. Yu, Z. Kuang, B. Zhang, W. Zhang, D. Lin, and J. Fan, "Leveraging content sensitiveness and user trustworthiness to recommend fine-grained privacy settings for social image sharing," IEEE Trans. on Information Forensics and Security, vol. 13, no. 5, pp. 1317–1332, May 2018.

[43] K. Tero and T. A. Samuli, Laine and, "A style-based generator architecture for generative adversarial networks," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Long Beach, CA, USA, June 2019, pp. 4401–4410.

[44] L. Tao and L. Lei, "AnonymousNet: Natural face de-identification with measurable privacy," CoRR, vol. abs/1904.12620, 2019. [Online]. Available: http://arxiv.org/abs/1904.12620

[45] D. Bau, H. Strobelt, W. Peebles, J. Wulff, B. Zhou, J.-Y. Zhu, and A. Torralba, "Semantic photo manipulation with a generative image prior," ACM Trans. Graph, vol. 38, no. 4, pp. 59:1–59:11, Jul. 2019.

[46] K. Chinomi, N. Nitta, Y. Ito, and N. Babaguchi, "PriSurv: Privacy protected video surveillance system using adaptive visual abstraction," in Proc. Int. Conf. on Advances in Multimedia Modeling, Kyoto, Japan, Jan. 2008, pp. 144–154.

[47] J. Wickramasuriya, M. Datt, S. Mehrotra, and N. Venkatasubramanian, "Privacy protecting data collection in media spaces," in Proc. Int. Conf. on Multimedia, New York, USA, Oct. 2004, pp. 48–55.

[48] P. Chriskos, O. Zoidi, A. Tefas, and I. Pitas, "De-identifying facial images using singular value decomposition and projections," Multimedia Tools and Applications, pp. 1–34, Nov. 2016.

[49] Á. Erdélyi, T. Winkler, and B. Rinner, "Privacy protection vs. utility in visual data," Multimedia Tools and Applications, pp. 1–28, Feb. 2017.

[50] F. Pittaluga and S. J. Koppal, "Pre-capture privacy for small vision sensors," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 39, no. 11, pp. 2215–2226, Nov. 2017.

[51] A. Oppenheim, A. Willsky, and S. Nawab, Signals & Systems (2nd Ed.). Upper Saddle River, USA: Prentice-Hall, Inc., 1996.

[52] T. Popkin, A. Cavallaro, and D. Hands, "Accurate and efficient method for smoothly space-variant Gaussian blurring," IEEE Trans. on Image Processing, vol. 19, no. 5, pp. 1362–1370, May 2010.

[53] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, Oct. 2007.

[54] A. Bas, W. A. P. Smith, T. Bolkart, and S. Wuhrer, "Fitting a 3D morphable model to edges: A comparison between hard and soft correspondences," in Proc. Asian Conf. on Computer Vision, Taipei, Taiwan, Nov. 2016, pp. 1–15.

[55] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Providence, USA, June 2012, pp. 2879–2886.

[56] D. E. King, "Dlib-ml: A machine learning toolkit," Journal of Machine Learning Research, vol. 10, pp. 1755–1758, July 2009.

[57] "Human Engineering Design Data Digest, Department of Defense Human Factors Engineering Technical Advisory Group," http://www.acq.osd.mil/rd/hptb/hfetag/products/documents/HE_Design_Data_Digest.pdf, pp. 80–82, Apr. 2000.

[58] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "OpenFace: A general-purpose face recognition library with mobile applications," CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., Jan. 2016.

[59] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Boston, USA, June 2015, pp. 815–823.

[60] L. Weiyang, W. Yandong, Y. Zhiding, L. Ming, R. Bhiksha, and S. Le, "SphereFace: Deep hypersphere embedding for face recognition," CoRR, vol. abs/1704.08063, 2017. [Online]. Available: http://arxiv.org/abs/1704.08063

[61] D. Jiankang, G. Jia, and Z. Stefanos, "ArcFace: Additive angular margin loss for deep face recognition," CoRR, vol. abs/1801.07698, 2018. [Online]. Available: http://arxiv.org/abs/1801.07698

[62] T. Nawaz and J. Ferryman, "An annotation-free method for evaluating privacy protection techniques in videos," in Proc. IEEE Int. Conf. on Advanced Video and Signal-based Surveillance, Karlsruhe, Germany, Aug. 2015, pp. 1–6.

[63] P. Chriskos, O. Zoidi, A. Tefas, and I. Pitas, "De-identifying facial images using projections on hyperspheres," in Proc. IEEE Int. Conf. and Workshops on Automatic Face and Gesture Recognition, vol. 04, Ljubljana, Slovenia, May 2015, pp. 1–6.

[64] O. Sarwar, A. Cavallaro, and B. Rinner, "Temporally smooth privacy-protected airborne videos," in Proc. IEEE Int. Conf. on Intelligent Robots and Systems, Madrid, Spain, Oct. 2018, pp. 1–6.