Achieving Anonymity Against Major Face Recognition Algorithms

Benedikt Driessen
Ruhr-University Bochum
benedikt.driessen@rub.de

Markus Dürmuth
Ruhr-University Bochum
markus.duermuth@rub.de

ABSTRACT

An ever-increasing number of personal photos is stored online. This trend can be problematic, because face recognition software can undermine user privacy in unexpected ways. Face de-identification aims to prevent automatic recognition of faces thus improving user privacy, but previous work alters the image in a way that makes them indistinguishable for both computers and humans, which prevents a widespread use.

We propose a method for de-identification of images that effectively prevents face recognition software (using the most popular and effective algorithms) from identifying people, but still allows human recognition.

We evaluate our method experimentally by adapting the CSU framework and using the FERET database. We show that we are able to achieve strong de-identification while maintaining reasonable image quality.

1. INTRODUCTION

The number of personal photos that is available online has been rapidly increasing over the past years.[1] This development is driven by the wide availability of (stationary and mobile) high-speed Internet, cheap electronic storage, and ubiquitous digital cameras on the one hand, and a strong trend towards social networks and managing friends online on the other hand. Recently, some services publicly announced the deployment of face recognition software on the stored images (see, e.g., [12]). Face recognition software can be beneficial for the user, as it helps finding and tagging friends in pictures. However, it can also be used to gather additional information about friendship-relations (i.e., the social graph), even relations the user deliberately did not share with everybody.

People often try to separate some groups of people, e.g., personal friends and work colleagues and reserve an online profile for personal friends. Note that Facebook considers the social graph as public information that can even be queried via a special API [13], and in general one can easily imagine external services crawling the image database. A rather drastic example includes predicting a persons sexual orientation from the social graph [22]. So a person might want to hide (parts of) their social graph in order to protect such information, but still might want to post images. (We stress that automated extraction of the social graph is a much bigger problem than manual extraction, because of the large-scale extraction of information that becomes possible.)

In the past few month, the criticism of automated face recognition, and in particular Facebook as the most prominent example, has increased, e.g., from the Electronic Privacy Information Center (EPIC), which considers Facebook's handling of personal data a violation of European privacy law (see, e.g., [35, 9]).

In this work we will demonstrate a system that effectively protects the anonymity, in particular the social graph, of a user by thwarting face recognition software, while still allowing humans to identify faces and keeping the visual changes to the image small. In particular, previous work [24] mapped several "similar" faces to the same "average"-face (see Section 2 for details), thus the resulting face-images are indistinguishable for both computers *and* humans. While their approach provides strong security guarantees, it constitutes a strong visual alteration of the face images, and in particular prevents humans from distinguishing persons. We believe that this is too intrusive to find wide-spread use.

1.1 Our contribution

We propose a system to anonymize face images in a way that retains more details of the original image than previous work, thus allowing a human to still identify the person from the image, and works against all major classes of face recognition algorithms.

We exploit the fact that face recognition algorithms reduce the dimensionality of the data of the face image in order to reduce noise and improve speed. We show how to manipulate these relevant parts of the image to fool all important face recognition algorithms. We aim at a slightly weaker form of anonymity than k-anonymity (see Section 4), where an attacker's confidence in correct identification is small. This is well suited for the two main threats we are considering: first, we want to prevent extraction of the relationships (the social graph), e.g., by evaluating who is present on a sufficiently large number of pictures of one specific person; second, we want to prevent automated tagging of people on pictures.

[1] As an example, Facebook hosted 10 billion photos in Oct. 2008, receiving about 250 million new photos a day [14, 11]. Flickr hosted 4 billion photos in Oct. 2009 [15].

Of course, when humans can recognize a face, then the face will not be anonymous in a strict sense. Studies show [2] that the "price" most users are willing to pay for privacy is pretty low, so we hope that by providing a reasonable image quality, this approach will find more acceptance by users, while still preventing automated extraction of information and thus providing a reasonable level of security.

1.2 Paper organization

We describe face recognition algorithms based on principle component analysis (Eigenfaces), namely PCA, PCA+LDA, and Bayesian classifiers, and explain the basic steps to fool these algorithms in Section 2. We review the Elastic Bunch Graph Mapping (EBGM) algorithm and the idea to fool these algorithms in Section 3. We describe our approach to anonymize images in Section 4, show a number of experiments in Section 5, and discuss these results in Section 6. We review related work in Section 7 and conclude with Section 8.

2. EIGENFACE-BASED FACE RECOGNITION

We introduce the basic terminology, face recognition algorithms based on Principal Component Analysis (PCA), and show the basic idea how we can manipulate these algorithms. We consider the classical Eigenface-algorithm [34], which is interesting because it constituted a breakthrough in recognition performance at the time of its invention, and still provides very competitive performance for images taken in a moderately controlled environment. Also, it forms the basis for a wide range of algorithms, including Linear Discriminant Analysis (LDA), which can be applied after PCA, and the Bayesian classifier, both of which we present in the sequel.

2.1 Brief introduction to face recognition terminology

The task of face recognition can be described in simple terms: Given a set of images of a number of (known) persons (*gallery images*), and given an image for an unknown person (*probe image*), decide which person from the gallery is shown on the probe image.

The process is usually divided into the following steps (see Figure 1): First, one needs to find the approximate position of the face in the image, this is called *face-detection* [38] and a separate line of research. Most work on face recognition considers this job to be completed before; commonly used face image databases such as the FERET database (see Section 5.2) annotate the images with the eye coordinates. Second, images are *normalized*, which usually includes an affine transformation to align the eyes, histogram equalization, and sometimes masking of the background. Third, in *feature extraction* algorithm-dependent features the probe image are extracted. Representing an image by a set of features can be seen as a step in data reduction that aims at extracting a compact but discriminating description of the image. Ideally, the output of this step is at the same time robust against changes in posture, lightning, face expression, etc. Finally, the pre-processed probe image is *matched* against gallery images. Different distance functions, measuring the similarity of two images, can be used here, we will see Euclidean distance, weighted Euclidean distance, and angles in the sequel.

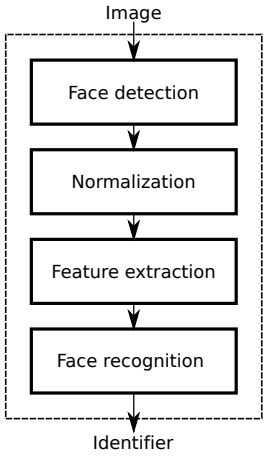

Figure 1: Process of identifying subjects in the real-world.

The output of a face recognition algorithm is a list of identifiers, where the algorithm estimates that the first identifier (e.g. name) is the most likely one, matching the subject on the probe image. The list is typically ordered by descending probability (or ascending distance, depending on the distance measure). The performance of face recognition algorithms is usually measured in *rank curves* (see Figure 2), where on the x-axis we plot ranks and on the y-axis recognition rates. When the curve passes point (x, y) this means that for a fraction of y probe images, the correct identifier was contained in the first x suggestions of the recognition algorithm.

2.2 Classical Eigenfaces

Given L training pictures $\vec{u_1}, \ldots, \vec{u_L}$, written as vectors with p pixels each, we compute the (point-wise) average image

$$\vec{m} := \frac{1}{L} \sum_{k=1}^{L} \vec{u_k}$$

and compute the mean-subtracted images $\vec{u'_k} := \vec{u_k} - \vec{m}$. Write $U = [\vec{u'_1}, \ldots, \vec{u'_L}]^t$ for the matrix of images. Then the (empirical) covariance matrix is written as

$$C = U^t \cdot U = \frac{1}{L} \sum_{k=1}^{L} \vec{u'_k} \vec{u'_k}^t.$$

The matrix U has size $p \times L$ and the matrix C has size $p \times p$. Let $\lambda_0, \ldots, \lambda_{N-1}$ be the N largest Eigenvalues with associated Eigenvectors $\vec{e}_0, \ldots, \vec{e}_{N-1}$.[2] Using the orthonormal vectors $\vec{e}_0, \ldots, \vec{e}_{N-1}$, we calculate a feature vector (in what is called *face-space*, the space spanned by Eigenvectors), by simply projecting an image $\vec{u}$ on the Eigenvectors

$$s_k = \vec{e_k}^t \cdot (\vec{u} - \vec{m}) \qquad \text{for } k = 0, \ldots, N-1, \tag{1}$$

[2]Computing the Eigenvectors of this large matrix is computationally expensive for common parameter sizes. Instead, one computes the Eigenvectors of the (usually smaller) matrix $U \cdot U^t$ and derives the Eigenvectors e_k from these.

where s_k is the k-th component of the projection of $\vec{u}$ in face-space. Calculating a feature vector for every gallery image as well as the probe image, and using Euclidean distance (where i iterates over the elements of the vectors),

$$D_{\text{Euclidean}}(\vec{u},\vec{v}) = \sqrt{\sum_i (\vec{u}^{(i)} - \vec{v}^{(i)})^2} \tag{2}$$

to find the gallery image that is closest to the probe image, we get the original Eigenfaces face recognition algorithm.

Instead of Euclidean distance we can also use different distance measures, such as MahCosine, which measures the angle of the projections $\vec{m},\vec{n}$ of two vectors $\vec{u},\vec{v}$ into Mahalinobis space, i.e.,

$$D_{\text{MahCosine}}(\vec{u},\vec{v}) = cos(\theta_{\vec{m},\vec{n}}), \tag{3}$$

see [4] for details and more examples. We essentially chose this measure because it outperformed other measures for purely PCA-based recognition methods [19] and in order to enhance our understanding of the effectiveness of de-identification methods in the presence of non-Euclidean distance measures.

2.3 Modifying projections (in image space)

Our basic idea is to manipulate the face images in such a way that the projection onto the face-space changes, while hopefully making minimal changes to the image.

Given the input image $\vec{u}$ as a row-vector and a set of orthogonal and normalized vectors $\vec{e_0},\ldots,\vec{e_{N-1}}$ (the selected principal eigenvectors as computed by the PCA, spanning the face-space), consider Equation (1). By adding $\Delta_k \cdot \vec{e_k}$, a multiple of the k-th Eigenvector, to an image $\vec{u}$, we can arbitrarily change the k-th component of the projection:

$$\begin{aligned}
&((\Delta_k \cdot \vec{e_k}^t + \vec{u}^t) - \vec{m}^t)) \cdot \vec{e_k} \\
&\quad = \Delta_k \cdot \vec{e_k}^t \cdot \vec{e_k} + (\vec{u}^t - \vec{m}^t) \cdot \vec{e_k} \\
&\quad = \Delta_k \cdot ||\vec{e_k}||_2^2 + s_k \\
&\quad = \Delta_k + s_k,
\end{aligned} \tag{4}$$

where we use that $||\vec{e_k}||_2^2 = 1$.

When adjusting several components, the projections behave independently, because the $\vec{e_1},\ldots,\vec{e_{N-1}}$ are pairwise orthogonal. Let

$$\vec{v} := \vec{u} + \sum_{l=0}^{N-1} \Delta_l \cdot \vec{e_l}$$

be the image we have modified based on the input image $\vec{u}$, then the projection of the altered image differs by Δ_k in the k-th component, i.e.,

$$\begin{aligned}
&(\vec{v}^t - \vec{m}^t) \cdot \vec{e_k} \\
&\quad = (\vec{u}^t + \sum_{l=0}^{N-1} \Delta_l \cdot \vec{e_l}^t - \vec{m}^t) \cdot \vec{e_k} \\
&\quad = (\vec{u}^t - \vec{m}^t) \cdot \vec{e_k} + \sum_{l=0}^{N-1} \Delta_l \cdot \vec{e_l}^t \cdot \vec{e_k} \\
&\quad = s_k + \Delta_k \qquad \text{for } k = 0,\ldots,N-1.
\end{aligned} \tag{5}$$

Compared with previous work [24], the image quality of this approach is better, because we leave the information outside the space spanned by the Eigenfaces untouched.

2.4 PCA+LDA face recognition

LDA can be applied directly to the input data [1], but was found to be more effective when applied after a PCA transform [42]. Writing E for the matrix that describes the PCA transform, LDA produces a matrix W that gives an optimal linear discriminant function, projecting the input into a classification space. W can be computed from the within-class and between-class scatter matrices; the details are not relevant for our work and we refer to [42]. For a combination of PCA and LDA, the overall transformation is from image space (via face-space) to classification space and can be written as

$$\vec{s} = W^t \cdot E^t \cdot \vec{u}, \tag{6}$$

where $\vec{s}$ is a vector in the classification space. In this space, a simple or weighted Euclidean distance such as the measure (again, k iterates over the elements of $\vec{u}$ and $\vec{v}$)

$$D_{\text{ldaSoft}}(\vec{u},\vec{v}) = \sum_k \lambda_i^{0.2} (\vec{u}_k - \vec{v}_k)^2,$$

proposed by Wen Yi Zao [40], can be used to find the nearest neighbor (where the λ_i are the LDA Eigenvalues).

As we can see, we do a normal PCA transform (albeit with a slightly different basis, as slightly different parameters are optimal for PCA with LDA) before applying LDA. Consequently, using the same techniques as in Section 2.3, we can alter the results for this method as well. In fact, even though the distance measures for purely PCA-based methods and this approach are different, de-identification for one system practically means de-identification for the other system as well, which can be understood from Equation (6).

2.5 Bayesian face recognition

The Bayesian face recognition algorithm [20, 33] is different from most other algorithms in that it breaks down face recognition to a series of classification problems: In order to recognize a face, the algorithm iterates over all stored *persons* (not faces), and for each decides if this is the correct person or not. The central idea is that it tries to decide if the difference of two faces is in one of two classes, either *inter-personal* (Ω_I) or *extra-personal* (Ω_E).

In the preprocessing stage, the algorithm learns what are "typical variations" for the difference of two images of the same face, and for two images of different faces. Consider the class Ω_I of inter-personal difference images, i.e., images of the form $\vec{\Delta} = \vec{u} - \vec{v}$, where $\vec{u},\vec{v}$ belong to the same person. In order to reduce the dimensionality of the data, one performs a PCA and keeps the M Eigenvectors with the largest Eigenvalues $\lambda_1^I,\ldots,\lambda_M^I$ and the projection matrix $E_I = (\vec{e_1^I},\ldots,\vec{e_M^I})$, which is used the to project the difference image Δ (with about 20 000 dimensions[3]) into the truncated intra-personal face-space (with about 300 dimensions). For the class Ω_E of extra-personal images with $\vec{\Delta} = \vec{u} - \vec{v}$, where $\vec{u},\vec{v}$ belong to different subjects, the same is done with a separate application of PCA. This yields eigenvectors $\vec{e_1^E},\ldots,\vec{e_M^E}$ which form the transformation matrix E_E for projections into the extra-personal face-space.

To estimate whether two images $\vec{u},\vec{v}$ are from the same subject, there are two variants of Bayesian face recognition that can be used: The Maximum a Posteriori classifier

[3]This depends on the size of the pre-processed images, which are 128×128 pixels in our case.

(MAP) computes the likelihood based on both intra- and extra-personal spaces, while the simpler Maximum Likelihood classifier (ML) bases its estimate on the intra-personal space only. For most applications, the two variants provide very similar results. The similarity measure in the Bayesian model is expressed as

$$S_{\text{Bayesian}}(\vec{\Delta}) = P(\vec{\Delta} \in \Omega_I) \quad \text{with} \quad \vec{\Delta} = \vec{u} - \vec{v}$$

which basically states the probability that the images $\vec{u}$ and $\vec{v}$ are from the set of intra-personal images with high probability (and from Ω_E with low probability), i.e., from the same person. We refer to [20, 33] for further details.

3. ELASTIC BUNCH GRAPH MAPPING FACE RECOGNITION

Elastic bunch graph mapping [36] is another algorithm for face recognition that fared very well in the FERET tests [31, 25]. What makes this algorithm particularly interesting is that it is fundamentally different from the previous algorithms: it is not based on PCA and is commonly classified as feature-based instead of holistic, i.e., it bases its decision on particular local features (eyes, mouth, ...) instead of a holistic view of the face.

3.1 Algorithm description

We give a brief overview of the algorithm, for more details we refer the reader to [36, 5]. A central tool for EBGM are *Gabor wavelets*, convolution kernels which are plane waves bounded by a Gaussian envelope function. Let ψ_j be a Gabor wavelet, then the convolution at point x with the image $\vec{u}$ is given by

$$J_j(x) = \int \vec{u}(x')\psi_j(x' - x)dx', \tag{7}$$

where $J_j(x)$ is a complex value and the index j ranges over 40 values for 8 orientations and 5 frequencies. Convolution of a fixed point of an image with Gabor wavelets of different orientation and frequency is called a *jet*; intuitively, a jet contains a reduced description of the surrounding of that point. Gabor wavelets are robust against a number of variations, and are motivated by human vision research.

Different distance functions for jets are used in different stages of the process, the most important one being the following: Writing the complex values as $J_j = a_j \cdot \exp(i\phi_j)$, the distance function is defined as

$$S_a(J, J') = \frac{\sum_j a_j a'_j}{\sqrt{\sum_j a_j^2 \sum_j {a'_j}^2}},$$

i.e., a "normalized vector product".

For the faces, one defines a set of *fiducial points*, such as pupils, corner of the eyes, corners of the mouth, and top and bottom of the ears. The nodes of these graphs are labeled with a jet. Initially, for a small set of faces, the fiducial points are extracted by hand, and the jets are computed.

When presented with a new face, the information extracted above is used to automatically fit the above graph to a new face: First, the rough position of the face is determined by matching the average of all above graphs onto the probe image. Then the graph that fits best is selected, allowing for small displacements and scaling of the graph, followed by successively relaxing the graph geometry and adapting the points individually.

The graph is fitted on every image in the gallery, and the resulting vector of jets is stored. For a probe image, the closest match with a gallery image is computed as the mean of the individual jet similarities:

$$\frac{1}{N} \sum_n S_a(J_n^I, J_n^G).$$

3.2 Modifying jets (in image space)

Our basic idea idea is that by adding appropriate multiples of a wavelet $\alpha \cdot \psi_j$ to an image $\vec{u}$ at position x, we can change the value of the convolution with this particular wavelet at the specific position:

$$\begin{aligned} J_j^*(x) &= \int (\vec{u}(x') + \alpha \cdot \psi_j(x' - x)) \cdot \psi_j(x' - x)dx' \\ &= \int \vec{u}(x') \cdot \psi_j(x' - x) + \alpha \int \psi_j(x' - x)^2 dx'. \end{aligned}$$

One difference to the situation for PCA is that these changes are not independent of each other: modifying one jet value also changes other values for this jet, and several jets are close enough that other jets are influenced as well. For this reason we proceed iteratively as follows:

1. Do 150 times, over all jets and wavelets:
   (a) Find the maximum difference between the current and target value
   (b) Add Gabor wavelet to bridge 1/5-th of the distance
2. Over all jets and wavelets (wavelets with large radius first):
   (a) Add Gabor wavelet to bridge 1/20-th of the distance

We established these parameters empirically and found them to work well. As for PCA-based techniques, it is not necessary to actually set the image to be equal to the target image, because probe and gallery images have a certain distance anyway.

4. ACHIEVING ANONYMITY

Next, we describe how we utilize the approaches from the previous sections to anonymize face images.

4.1 k-anonymity

An established definition of security against identification is *k-anonymity* [32], see also the notion of *anonymity sets* [26]. For face recognition, a person remains k-anonymous if the face recognition algorithms cannot narrow the person down to a set of less then k persons.

For our envisioned targets, weaker forms of anonymity are sufficient. There are two scenarios we would like to protect against: First, automated tagging of persons on uploaded pictures (note that we are not targeting the automated proposal of persons to tag, because this still contains human interaction and thus is only making tagging easier...); second, automated derivation of friendship-relations from a large set of pictures. For both scenarios, weaker privacy guarantees suffice. Along with this weaker privacy guarantee comes a

large improvement in image quality (as perceived by a human), so we hope that our system will lead to more widespread usage of privacy-protecting systems.

4.2 Anonymizing face images

Here we describe our approach to face anonymization, which builds on the methods we have developed above.

1. We select a partition of the involved persons such that each set has at least k members. We choose them by picking a random face image and selecting the $k-1$ nearest images (of distinct persons) according to a suitable distance measure (we will elaborate on this in Section 5) and we call this set a *cluster*.

2. For every cluster we project each of the k images and compute the average projection (wrt. to PCA).

3. All images in a cluster are modified (as described in Section 2.3) to have the same, averaged projection. However, we may also choose to adjust the projection by a fraction $\sigma \in \mathbb{R}$ with $0 < \sigma < 1$ of the difference between an images actual projection and the average projection. The parameter σ was determined experimentally, see Section 5.4.

4. Next, all images in a cluster are modified wrt. to EBGM (as described in Section 3.2) to resemble the average face for that cluster. We apply EBGM-modifications after PCA-modifications, because EBGM-modifications are local, whereas PCA-changes influence the entire image (see classification of EBGM as a local feature-approach, as opposed to PCA being a holistic approach).

A central observation is that we do not need to change the projections of PCA in face-space to the actual average, but it's sufficient to go some way in that direction. The reason is that probe image and gallery image of the same person are already quite some distant apart, so moving partially in the correct direction suffices. We will show experiments substantiating this claim.

5. EXPERIMENTS

In this section we present extensive experiments that substantiate our privacy claims. We used the CSU framework of face recognition algorithms to test our results on a subset of 1000 images of the FERET database. We performed de-identification experiments for all three classes of algorithms (Eigenface-based, Bayesian, EBGM) and finally realized a synthesis of our results.

5.1 The CSU framework

The CSU framework [6, 4] was created at Colorado State University to facilitate the comparison of different algorithms, and is available for free for research. The framework runs on UNIX/Linux systems, the source code is available and therefore easily adaptable. The current version 5.1, published July 2010, supports the following face recognition algorithms:

- Classical Eigenfaces (i.e., PCA) with different distance measures, e.g. Euclidean, MahCosine, etc.,
- LDA+PCA, also with different measures,
- Bayesian classification with the MAP and ML classifier, and
- Elastic Bunch Graph Mapping.

Details about the specific implementations can be found in a series of papers, most notably [33, 5, 37, 3]. The framework utilizes the FERET dataset and allows to easily measure the recognition performance. Furthermore, due to its modularity, the framework can be extended by new algorithms in order to benchmark these against already known methods. Although not intended, the framework can easily be adapted to allow benchmarking the de-identification performance of our algorithms.

5.2 The FERET database

The FERET program [27, 29] started 1993 and ran until 1997. It was sponsored by the Department of Defense Counterdrug Technology Development Program through the Defense Advanced Research Products Agency. Its primary mission was to develop automatic face recognition algorithms that could be employed to assist security, intelligence and law enforcement personnel. The FERET dataset was assembled to support government monitored testing and evaluation of face recognition algorithms using standardized tests and procedures. The final set of images has 3300 images from 1200 persons, with varying mimical expressions, from different dates, under semi-controlled conditions. The dataset is available for for research related to face recognition.

5.3 Scope and conduct of experiments

For our experiments, we used the FA and FB subsets of the FERET database, each containing one facial expression of 1195 subjects. We ran the experiments on a random subsets of 500 subjects each. The FA set served as our gallery, face recognition performance figures are given in terms of successively matching subjects from the probe set FB to their alternate image in FA. In our experiments, we apply our de-identification methods against both sets of images which implies that, although all gallery images have been de-identified, the face recognition system still has identifiers associated to them. This is not only a necessary pre-requisite for identification, but also realistic when considering that most social networks encourage users to manually identify persons on photos, thus adding images to the gallery. Several experiments were performed to validate our de-identification approaches and identify a reasonable set of parameters. First, we tested our de-identification method for each recognition algorithm individually; the results are shown in Section 5.4 and Section 5.5. Then we combined these preliminary findings, applying both techniques at the same time, for our final results in Section 5.6.

All experiments were conducted with the default configuration of the CSU framework, the only exception concerns the normalization step. EBGM uses a pre-processing procedure which is different from the normalization required by all other recognition methods. Our experiments were conducted entirely on the data-set generated by EBGM pre-processing. The reason is our two-fold strategy for de-identification. Ideally, all PCA-related modifications are applied to images normalized for PCA-based methods, then the result is transformed back to the original image which is subsequently pre-processed for EBGM. Finally, the EBGM

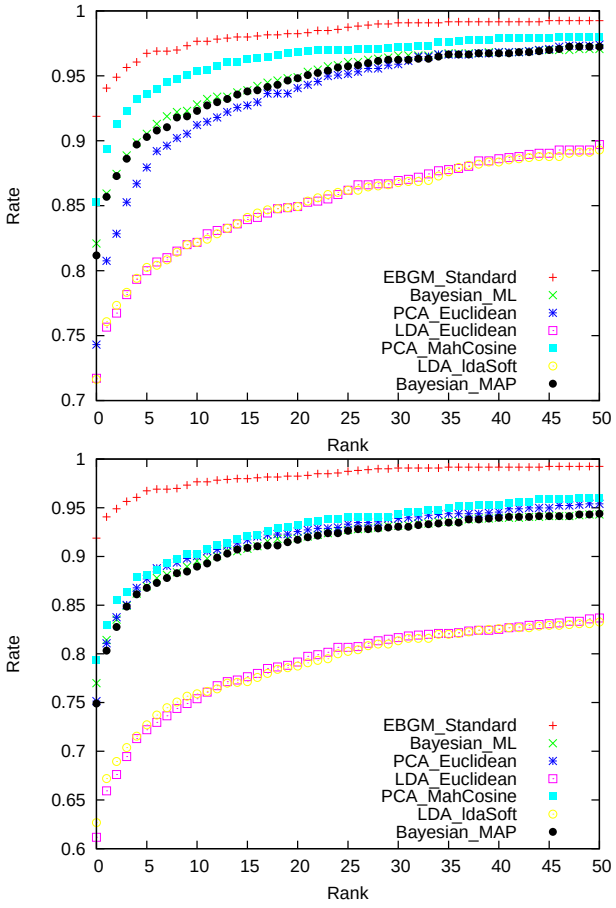

Figure 2: Baseline performances of the algorithms (prior to de-identification), in the first image with the original normalization parameters, in the second image with identical normalization for all algorithms.

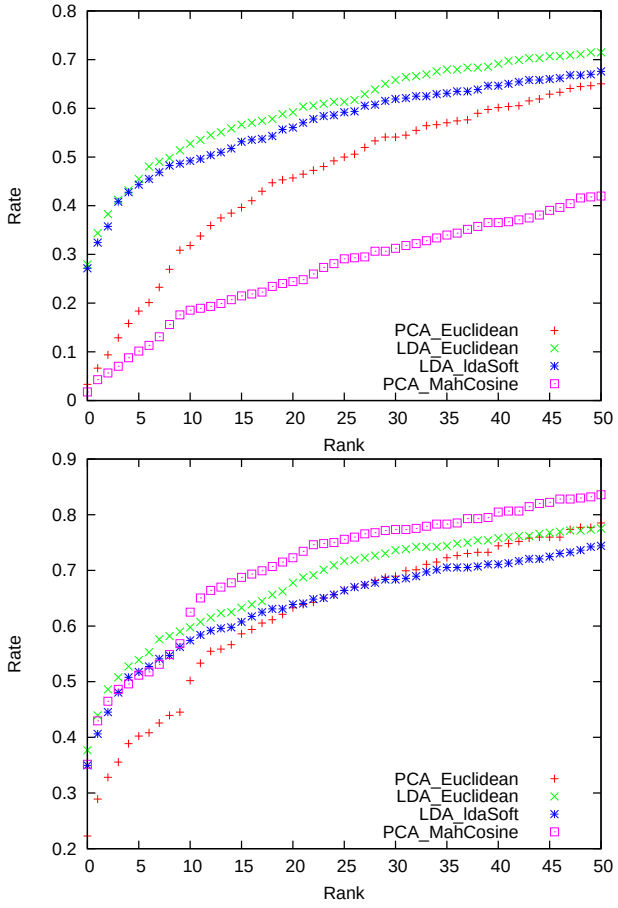

Figure 3: Eigenface-based recognition after de-identification with $k = 10$ and $\sigma \in \{1, 0.85\}$ clusters obtained from the same measure.

modifications are applied. However, this is tedious work that adds little insight to the interesting questions, so we chose to use the EBGM data-set for all methods. The impact of this strategy on recognition rates is shown in Figure 2. It displays the rank curves for all of the algorithms targeted by us. The upper graph shows recognition performance (prior to de-identification) in the default configuration, i.e. where each class of algorithms operated on specifically normalized images. The lower graph shows a slight degradation in recognition performance, which is due to non-optimal normalization, which seems to affect LDA-based methods most. The performance results of our de-identification methods, which are expressed as rank curves as well, are to be understood in relation to the lower graph.

5.4 Experiments for Eigenface-based face recognition

Experiments to determine the effectiveness of our de-identification method against Eigenface-based methods are parameterized by k, which is the size of the anonymity clusters, and $0 < \sigma < 1$ which is a factor weighting the addition of modifications. In a first series of experiments, presented in this section, we tested and validated our de-identification method against face recognition using PCA and LDA+PCA. For both, we tested different distance measures, which gives us assurance that the proposed method is robust against changes in this metric.

In a first series of experiments, we targeted each of these methods individually by building clusters of persons according to the same face recognition method, because we wanted to learn how sensitive our method is to how exactly the clusters are chosen. For our first experiment we consider de-identification with $\sigma = 1, k = 10$ and measure the recognition rate of all four methods, again. Comparing the upper graph in Figure 3 with the original results from Figure 2, we see that the two PCA algorithms perform quite similar with an extremely low rank-0 recognition rate, and the the PCA+LDA algorithms perform better, yet still much lower than without de-identification. Also, we can see that the performance of the MahCosine distance measure, which is very accurate without de-identification, decreases disproportionately strong.

In a second series of experiments, we determined a suitable weighting factor σ. The lower we choose σ, the less an image is actually altered (thus not completely bridging the distance to the cluster's target image) which consequently yields a better image quality. We found that de-identification with $k = 10$ and $\sigma = 0.85$ works well, and

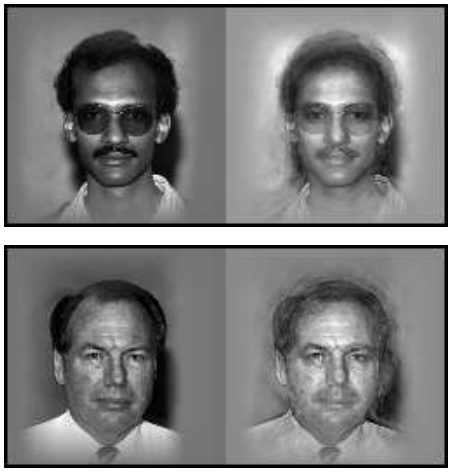

Figure 4: Comparison of original image (left) with image after de-identification (right) with parameters $k = 10, \sigma = 0.85$.

additionally this parameter balances the recognition rate for the four algorithms, as can be seen in the lower graph in Figure 3. Figure 4 shows the visual effects of de-identification for $k = 10, \sigma = 0.85$ for all four recognition methods in comparison with the original image. In both cases the person is clearly recognizable. The strongest effect on the pictures can be seen at the line between the person's hair and the background. This effect hardly affects the recognizability of a face, and can most likely be avoided by restricting the Eigenfaces to the actual face, as in the usual preprocessing for PCA. Also, the modified images look somewhat lighter than the original images.

In the previous experiments, we have selected the image clusters using the same distance measure as in the recognition task. This is not a realistic option in practice, so in a third series of experiments, we determined the best approach to de-identify images using a single clustering. We tested two approaches:

- Compute the four de-identified images for one subject, each grouped by one of the Eigenface-based measures, and average these pixel-wise.
- Compute clusters using a single distance measure, here we used the MahCosine measure (which performed best for $\sigma = 0.85$).

The results for both approaches are shown in Figure 5. The graph on the top shows that averaging over the four de-identified images per subject yields rather bad results, all algorithms have a rank-0 recognition rate of 40%-55%. The likely reason for this is that the clusters used to compute each of these projections where different, and the right image lies in the intersection between these. The lower graph shows much better de-identification and is our preferred method. The recognition rate of the classical Eigenfaces method is worse than in the case were we *specifically* targeted this method, see Figure 3. The curves of all other three algorithms very much resemble the already known results as in the afore mentioned graph. What is more important, they are still worse than the performance of the MahCosine distance measure, which serves as our benchmark in this case.

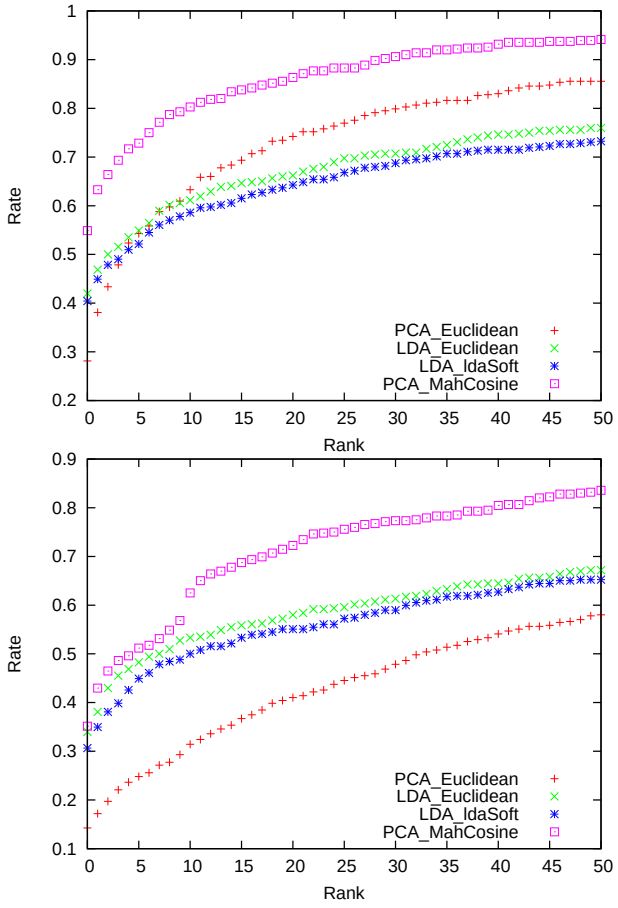

Figure 5: Eigenface-based recognition after de-identification of the same cluster with $k = 10, \sigma = 0.85$, using averaging (top) and MahCosine (bottom).

We conclude that clustering wrt. MahCosine achieves a high degree of de-identification among all tested Eigenface-based methods while only minimally impacting identification by humans. We expect that other distance measures exhibit similar performance.

5.5 Experiments for Bayesian face recognition

For the Bayesian face recognition, there are two (related) classifiers: MAP operates on both intra- and extra-personal spaces, while the simpler ML classifier bases its estimate on the intra-personal space only. For most applications, the two variants provide very similar results, but our first set of experiments targets each method individually.

For the Bayesian method we have again performed de-identification as described previously. We have grouped $k = 10$ subjects into clusters, determined by their closeness according to the classification by MAP and ML. Then we have de-identified the clusters with $\sigma = 0.85$. The performance of the recognition algorithms is shown in Figure 6 and is in line with the results expected due to prior experiments for Eigenface-based algorithms.

Figure 7 shows the visual effects of the de-identification. Again we see that both subjects are clearly recognizable by

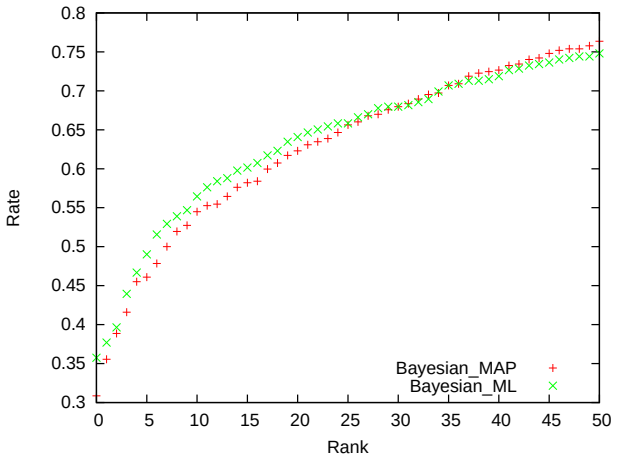

Figure 6: Bayesian recognition after de-identification with $\sigma = 0.85$ and $k = 10$.

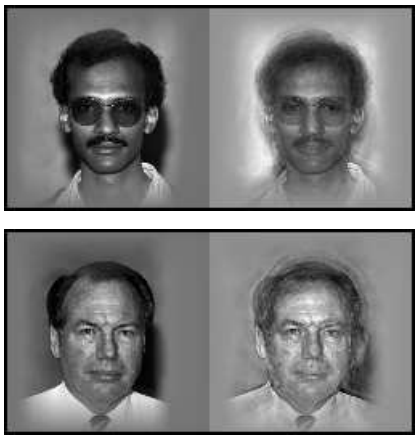

Figure 7: Comparison of image quality after de-identification with $k = 10, \sigma = 0.85$ for clusters grouped by MAP and ML.

humans, although the outline of the heads is blurred to a certain degree, which we attribute to variations in the preprocessing/normalization of the EBGM-method as we explained before.

5.6 Experiments for combined face recognition

Finally, we combined the method of modifying PCA-based projections with our algorithm to alter EBGM-specific features. Since clustering subjects by their closeness according to MahCosine proved to be effective against Eigenface-based approaches, we decided to do the same for this experiment. We have de-identified the probe- and gallery-set with $k = 10$ and $\sigma = 0.85$ in the first step and then modified the jets of the resulting images to resemble the average of the same set of clusters (still grouped by MahCosine).

Figure 8 shows the recognition performance of all algorithms when operating on the same set of images. We see that the EBGM algorithm fares better in recognizing de-identified subjects than the other methods, but still only gets a 55% rank-0 recognition rate, see the discussion in the next section. Interestingly, the curves for Bayesian MAP, ML and

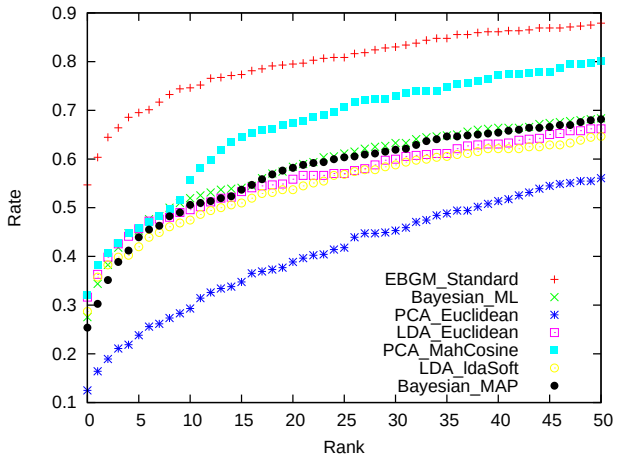

Figure 8: Recognition rates of all algorithms after de-identification with $\sigma = 0.85$ and $k = 10$.

Eigenface-based methods are very close to each other.

Figure 9 shows the visual outcome of the de-identification procedure. The images are all taken from the same cluster (of 10 images total). Note that previous work (e.g., [17]) would have assigned the *same* average image to all of these images, i.e., they would be indistinguishable for humans. However, the images produced by our method are clearly distinguishable, which was the main goal of our work.

6. DISCUSSION OF RESULTS

Face recognition algorithms work very well for images taken in a controlled environment (e.g., background, illumination, tilt). For example, the CSU implementation of the EBGM algorithm achieves a rank-0 recognition rate of more than 90% (c.f. Figure 8), and the original EBGM implementation fared even better. Our modifications reduce the rank-0 recognition rate to 55% for EBGM, and below 30% for the other algorithms, which is a big improvement.

The EBGM algorithm is, according to our experiments, harder to fool than other algorithms. Possible reasons are that EBGM is a feature-based algorithm (i.e., it works on small patches of the face, not a holistic view of the face), and it has a very good recognition rate in general. In practice, images stored on image sharing sites are not taken in a controlled environment, and we expect that in a real environment the recognition rates will drop substantially compared to the above experiments.

Sometimes, a corporation's interest in collecting data and a user's interest in privacy are diametrical. In the scenario we consider the corporation's interest is extracting as much information as possible from the available data, where some users want this information to be private. We stress again that there is a big difference between extracting such information automatically, or being "in principle" able to extract this information, but requiring human assistance for the task: Whereas the former scales well to large databases, the latter quickly becomes infeasible.

That said, using anonymization techniques will probably cause a reaction by the corporations deploying face recognition algorithms. They could try to detect modified images and lock them out, which would discourage users from using such techniques. It is not clear if the modifications of our

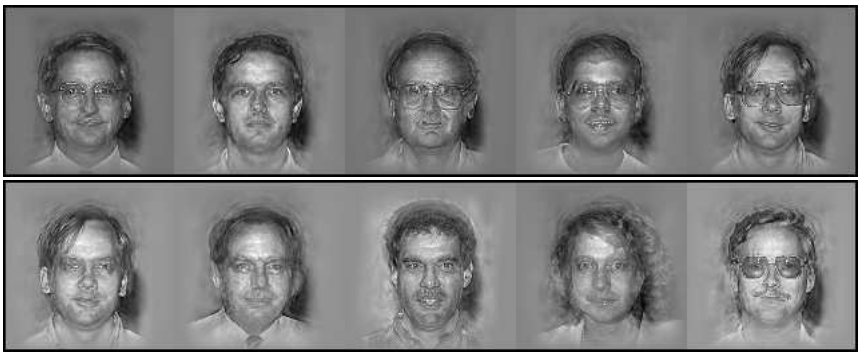

Figure 9: Image quality after de-identification for all algorithms with $k = 10, \sigma = 0.85$. The images shown represent one cluster grouped according to MahCosine, i.e., previous work maps these to the *same* average image.

method can be detected reliably, as the wide variations in real-world images make this task pretty hard. This would most likely lead to an arms-race between corporations and privacy advocates. Still, we believe that detecting the modifications can be made hard enough for practical purposes. Future research is required to solve some of the remaining issues, as well as to make it fully practical. But we are optimistic that we made a step in the right direction, and that the remaining problems can be solved.

7. RELATED WORK

7.1 Previous work on de-identification

Systematic research on face de-identification started with work by Newton, Sweeney, and Malin (e.g., [23, 24]). Their goal was to achieve *perfect anonymity*, which makes identifying an individual from the anonymized photo provably impossible for machines (and humans). Their first approach, called "k-same"-method, computed the de-identified image either averaging the closest k images pixel-wise ("k-same-pixel"-method) or in Eigenface ("k-same-eigen"-method). While achieving strong security guarantees, the resulting image quality of both approaches was mediocre. To improve the visual quality of the anonymized pictures, a refined method [17] uses Active Appearance Models (AAMs) fit to the k images in a cluster and averages over the parameters of the respective representations in the chosen model. Subsequent work [16] introduces a framework and defines different notions of privacy protection models. Putting it brief, our approach is to sacrifice some of the strong security guarantees this line of work gives, but giving humans a chance to recognize people on the images. We are convinced that this is a crucial step towards getting face de-identification accepted for use in practice, as the purpose of publishing images is often subverted if one cannot recognize the people on the image.

Another potential problem of the original proposal [23] (and potentially the newer works as well) appears when applied in practice (which we have not found in the literature). In order to explain our concerns, we briefly re-call the k-same method: In a *setup step*, one computes clusters of k similar faces and an average image that is later used as a substitute for this cluster. One starts with a set U that contains one picture for every person (a person-specific face set in their language). One random picture $\vec{u} \in U$ is selected, and the $k-1$ closest faces together with $\vec{u}$ form form a set V. The images in V are removed from U, the average image $\vec{v}$ is computed.

When multiple images of the same person are de-identified, we first need to decide which person is on the image (i.e., run a face recognition algorithm) in order to find the correct cluster and the corresponding average image. However, if an image is mis-classified for person B, but really is an image of person A, then different images of the same person will be sent to different clusters and hence assigned distinct average images. In this case k-anonymity is no longer guaranteed to hold, as the correct person lies in the intersection of possibly multiple sets. Experiments based on the Eigenfaces implementation in the CSU framework with standard pre-processing show this happens in the FERET database for multiple images, we found `00357fb010_940422.sfi` which is is identified as `00933fb010_960627.sfi`, which is not in the cluster created by grouping images for `00778fa010_941205.sfi`. Several more examples of this can be found, depending on the order in which images are selected from the image sets. Our method of selecting clusters is similar to the above, so our approach suffers from the same problem, however, we do not aim for perfect anonymity anyway.

A couple of ad-hoc methods such as masking parts of the face (e.g. the eyes) or blurring or pixelation of faces [7, 18, 21, 39] were eventually tested. However, these methods are visually intrusive and target human and algorithmic recognition alike. Even worse, it was shown that their effectiveness is very limited [24], so they are not a good option. Pixelation and scrambling techniques for videos can be found in [28, 8].

7.2 More related work

The area of face recognition techniques (and related disciplines such as face detection) has been an active area of research for the past 20 years. Reasons include numerous applications, ranging from access control to border protection (recently adding automated crawling of large image databases), as well as the increase of available computing power. The literature on face recognition is vast, see [41] for an overview. Probably the most influential paper was face recognition based on Eigenfaces [34], which applied principal component analysis for the task, which was refined in a number of ways since then. LDA, which can itself be used for

face recognition [10, 1], has been combined with PCA [42]; and PCA forms the basis of Bayesian face recognition [20].

A consequence of the high relevance of accurate recognition technology was the large, DARPA-funded FERET study [31, 27]. The image database collected and used in this study is one of the biggest databases available for research. Another surge in research activity was followed the Face Recognition Grand Challenge [30] in 2004, this time sponsored by IARPA and DHS. The focus of this challenge was improving algorithmic performance for progressively difficult high-resolution[4] 2D and 3D images.

While face recognition has many beneficial applications, it can be a threat to user privacy when applied on a large scale, e.g., by crawling large image databases which possibly allows for the automated extraction of possibly sensitive information. However, research on de-identification methods to maintain privacy in the presence of automated face recognition is scarce.

8. CONCLUSION

We have shown a reliable way to de-identify face images against a wide range of currently available recognition algorithms. While not achieving the very strong notion of k-anonymity, we achieved a level of anonymity which is sufficient to counter the two most pressing problems that face recognition software poses for users of social networks: First, automated extraction of the social graph, i.e, friendship relations; second, automated tagging of people in images. At the same time we get good image quality, which means that humans still can identify the person in the image. (This was not the case in previous work that achieves k-anonymity.)

We believe that it is possible to further increase the image quality, in particular to remove the ghosting artifacts surrounding the heads. We hope to be able to replace the iterative algorithm for EBGM modifications by faster code. Also interesting is the integration of our approach in a full tool that performs all operations on images of arbitrary posture, not just frontal images as used in the FA and FB sets, as well as integrating face de-identification into web-browsers for ease of use.

Acknowledgment

Portions of the research in this paper use the FERET database of facial images collected under the FERET program, sponsored by the DOD Counterdrug Technology Development Program Office [27, 29].

9. REFERENCES

[1] Belhumeur, P., Hespanha, J., and Kriegman, D. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. In *Proc. of the 4th European Conference on Computer Vision, ECCV'96* (1996), pp. 45–58.

[2] Beresford, A. R., Kübler, D., and Preibusch, S. Unwillingness to pay for privacy: A field experiment. IZA Discussion Paper No. 5017, available at `http://ftp.iza.org/dp5017.pdf`, 2010.

[4] In this paper we do not assume that 2D/3D high-resolution images are available but rather focus on 2D images of medium quality.

[3] Beveridge, J., and She, K. Fall 2001 update to the CSU PCA versus PCA+LDA comparison. Available at `http://www.cs.colostate.edu/evalfacerec/`, 2001.

[4] Beveridge, R., Bolme, D., Teixeira, M., and Draper, B. The CSU face identification evaluation system user's guide: Version 5.0. Online at `http://www.cs.colostate.edu/evalfacerec/`, 2003.

[5] Bolme, D. S. Elastic bunch graph matching. Master's thesis, CSU Computer Science Department, 2003.

[6] Bolme, D. S., Beveridge, J. R., Teixeira, M., and Draper, B. A. The CSU face identification evaluation system: Its purpose, features and structure. In *Proc. International Conference on Vision Systems* (2003), pp. 304–311.

[7] Boyle, M., Edwards, C., and Greenberg, S. The effects of filtered video on awareness and privacy. In *Proc. of the 2000 ACM conference on Computer supported cooperative work* (2000), CSCW '00, ACM, pp. 1–10.

[8] Dufaux, F., and Ebrahimi, T. Scrambling for Video Surveillance with Privacy. In *IEEE Workshop on Privacy Research in Vision* (2006).

[9] EPIC.org. EPIC files complaint, urges investigation of Facebook's facial recognition techniques. Online at `http://epic.org/2011/06/epic-files-complaint-urges-inv.html`, 2011.

[10] Etemad, K., and Chellappa, R. Discriminant analysis for recognition of human face images. *Journal of the Optical Society of America 14*, 8 (1997), 1724–1733.

[11] Facebook Announcement: 10 billion photos. Online at `http://www.facebook.com/note.php?note_id=30695603919`. Accessed 27.9.2011.

[12] Facebook Announcement: Making Photo Tagging Easier. Online at `http://www.facebook.com/blog.php?post=467145887130`. Accessed 27.9.2011.

[13] Facebook Inc. Online at `http://developers.facebook.com/docs/opengraph/`.

[14] Facebook Statistics page. Online at `http://www.facebook.com/press/info.php?statistics`. Accessed 27.9.2011.

[15] Flickr Blog: 4 billion photos. Online at `http://blog.flickr.net/en/2009/10/12/4000000000/`. Accessed 27.9.2011.

[16] Gross, R., and Sweeney, L. Towards real-world face de-identification. In *Biometrics: Theory, Applications, and Systems, 2007. BTAS 2007. First IEEE International Conference on* (2007), pp. 1 –8.

[17] Gross, R., Sweeney, L., de la Torre, F., and Baker, S. Model-based face de-identification. In *Proc. of the 2006 Conference on Computer Vision and Pattern Recognition Workshop* (2006), CVPRW '06, IEEE Computer Society.

[18] Hudson, S. E., and Smith, I. Techniques for addressing fundamental privacy and disruption tradeoffs in awareness support systems. In *Proc. of the 1996 ACM conference on Computer supported cooperative work* (1996), CSCW '96, ACM, pp. 248–257.

[19] Miller, P., and Lyle, J. The effect of distance measures on the recognition rates of PCA and LDA based facial recognition. Tech. rep., Clemson University, 2008.

[20] Moghaddam, B., Jebara, T., and Pentland, A. Bayesian face recognition. *Pattern Recognition 33*, 11 (2000), 1771–1782.

[21] Neustaedter, C., Greenberg, S., and Boyle, M. Blur filtration fails to preserve privacy for home-based video conferencing. *ACM Trans. Comput.-Hum. Interact. 13* (March 2006), 1–36.

[22] New York Times. How privacy vanishes online. Available online at `http://www.nytimes.com/2010/03/17/technology/17privacy.html`, 2010.

[23] Newton, E., Sweeney, L., and Malin, B. Preserving privacy by de-identifying facial images. Tech. Rep. CMU-CS-03-119, Carnegie Mellon University, School of Computer Science, 2003.

[24] Newton, E., Sweeney, L., and Malin, B. Preserving privacy by de-identifying facial images. *IEEE Transactions on Knowledge and Data Engineering 17*, 2 (2005), 232–243.

[25] Okada, K., Steffens, J., Maurer, T., Hong, H., Elagin, E., Neven, H., and von der Malsburg, C. The Bochum/USC face recognition system and how it fared in the FERET phase III test. In *Face Recognition: From Theory to Applications* (1998), pp. 186–205.

[26] Pfitzmann, A., and Köhntopp, M. Anonymity, unobservability, and pseudonymity – a proposal for terminology. In *Workshop on Design Issues in Anonymity and Unobservability* (2000), pp. 1–9.

[27] Phillips, J. P., Moon, H., Rizvi, S. A., and Rauss, P. J. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 10 (2000), 1090–1104.

[28] Phillips, P. Privacy operating characteristic for privacy protection in surveillance applications. In *Audio- and Video-Based Biometric Person Authentication*, vol. 3546 of *LNCS*. Springer, 2005, pp. 869–878.

[29] Phillips, P., Wechsler, H., Huang, J., and Rauss, P. J. The FERET database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing 16*, 5 (1998), 295 – 306.

[30] Phillips, P. J., Flynn, P. J., Scruggs, T., Bowyer, K. W., Chang, J., Hoffman, K., Marques, J., Min, J., and Worek, W. Overview of the face recognition grand challenge. In *Proc. of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01* (2005), CVPR '05, IEEE Computer Society, pp. 947–954.

[31] Phillips, P. J., Rauss, P. J., and Der, S. Z. FERET (face recognition technology) recognition algorithm development and test results. Tech. Rep. ARL-TR-995, Army Research Laboratory, 1996.

[32] Sweeney, L. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 10*, 5 (2002), 557–570.

[33] Teixeira, M. L. The Bayesian intrapersonal/extrapersonal classifier. Master's thesis, Colorado State University, 2003.

[34] Turk, M., and Pentland, A. Eigenfaces for recognition. *Journal of Cognitive Neuroscience 3*, 1 (1991), 71–86.

[35] Welle, D. Facebook facial recognition raises eyebrows in Germany, EU. Online at `http://www.dw-world.de/dw/article/0,,15144128,00.html`, accessed Nov 14, 2011.

[36] Wiskott, L., J.-M., F., Kruger, N., and Malsburg, C. V. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*, 7 (1997), 775–779.

[37] Yambor, W., Draper, B., and Beveridge, R. Analyzing PCA-based face recognition algorithms: Eigenvector selection and distance measures. In *Empirical Evaluation Methods in Computer Vision* (2002).

[38] Yang, M.-H., Kriegman, D., and Ahuja, N. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 1 (2002), 34–58.

[39] Zhao, Q. A., and Stasko, J. T. Evaluating image filtering based techniques in media space applications. In *Proc. of the 1998 ACM conference on Computer supported cooperative work* (1998), CSCW '98, ACM, pp. 11–18.

[40] Zhao, W., Chellappa, R., and Phillips, P. Subspace linear discriminant analysis for face recognition. Tech. rep., UMD–TR4009, 1999.

[41] Zhao, W., Chellappa, R., Rosenfeld, A., and Phillips, P. Face recognition: A literature survey. *ACM Computing Surveys* (2003), 399–458.

[42] Zhao, W., Krishnaswamy, A., Chellappa, R., Sweets, D., and Weng, J. Discriminant analysis of principal components for face recognition. In *Proc. of the 3rd IEEE International Conference on Face and Gesture Recognition* (1998), pp. 336–341.