Preserving privacy by de-identifying face images

Article *in* IEEE Transactions on Knowledge and Data Engineering · March 2005

DOI: 10.1109/TKDE.2005.32 · Source: IEEE Xplore

3 authors, including:

Latanya Sweeney
Harvard University
60 PUBLICATIONS **5,039** CITATIONS

SEE PROFILE

Bradley Malin
Vanderbilt University
218 PUBLICATIONS **6,584** CITATIONS

SEE PROFILE

Preserving Privacy by De-Identifying Face Images

Elaine M. Newton, *Student Member*, *IEEE*, Latanya Sweeney, *Member*, *IEEE Computer Society*, and Bradley Malin, *Student Member*, *IEEE*

Abstract—In the context of sharing video surveillance data, a significant threat to privacy is face recognition software, which can automatically identify known people, such as from a database of drivers' license photos, and thereby track people regardless of suspicion. This paper introduces an algorithm to protect the privacy of individuals in video surveillance data by de-identifying faces such that many facial characteristics remain but the face cannot be reliably recognized. A trivial solution to de-identifying faces involves blacking out each face. This thwarts any possible face recognition, but because all facial details are obscured, the result is of limited use. Many ad hoc attempts, such as covering eyes, fail to thwart face recognition because of the robustness of face recognition methods. This paper presents a new privacy-enabling algorithm, named *k*-Same, that guarantees face recognition software cannot reliably recognize de-identified faces, even though many facial details are preserved. The algorithm determines similarity between faces based on a distance metric and creates new faces by averaging image components, which may be the original image pixels (*k*-Same-Pixel) or eigenvectors (*k*-Same-Eigen). Results are presented on a standard collection of real face images with varying *k*.

Index Terms—Video surveillance, privacy, privacy-preserving data mining, k-anonymity.

1 Introduction

There has been a tremendous proliferation of video surveillance cameras in public locations such as stores, ATMs, schools, subway stations, and airports. For example, a recent survey of Times Square found 500 visible surveillance cameras in the area and 2,500 total in New York City [8]. The existence of so many cameras and proposals for even more cameras have caused many citizens to protest their use in Tampa, Florida, and Virginia Beach, Virginia, where video cameras are being used in conjunction with face recognition software [9], [16], thereby enabling the eventual possibility of tracking almost all of the people most of the time.

One could imagine in the near future the widespread use of video surveillance cameras, the subsequent sharing of that data, and the use of reliable face recognition software to explicitly identify and track people in real time. This extends beyond American society's current expectations, thereby challenging a person's "reasonable expectation of privacy" as provided in US law. Yet, there are many worthy purposes for sharing video surveillance data. Three examples include law enforcement, bioterrorism surveillance, and medical research.

In the case of law enforcement, obtaining (unaltered) video surveillance footage from its owner may require a search warrant, particularly if the footage was not recorded in a public space. Alternatively, if the subjects in the footage could be "anonymized," then authorities could view the tape beforehand to gain sufficient evidence for a search warrant as warranted [6]. A very specific search warrant could be issued to reveal the identity of an actor exhibiting suspicious behavior or possibly a witness who might have been looking in the direction of a crime taking place. Therefore, the technology described in this work could allow the general sharing of the tapes with law enforcement, while still providing traditional privacy protections to those who are not under suspicion.

Second, government agencies engaged in bioterrorism surveillance want to be able to observe signs of respiratory distress (e.g., coughing) in the general population to discover early evidence of an outbreak or a bioterrorism attack [22]. By observing anonymized video surveillance images, this work allows for the general sharing of video tapes originally captured for safety and security without revealing the identities of the individuals, in accordance with existing public health laws and to allay fears in the commercial sector who may supply video footage.

Third, using video data as a basis for conducting behavioral research is becoming quite common (e.g., Alzheimer's patients living in a specialized care unit [3]). Because of the restrictions of the Health Information Portability and Accountability Act (HIPAA), this type of medical research is limited in its ability to widely share its recordings unless technology as described in this work can be used to reasonably anonymize the images. The identity of nonconsenting patients as well as employees would need to be concealed.

The goal of this work is to enable the sharing of video data with scientific assurances of privacy protection while keeping the data practically useful. Experiments reported herein demonstrate that many tactics done on television,

- *E.M. Newton is with the Department of Engineering and Public Policy, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213-3890. E-mail: enewton@andrew.cmu.edu.*
- *L. Sweeney and B. Malin are with the Data Privacy Laboratory, School of Computer Science, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213-3890. E-mail: {latanya, malin}@andrew.cmu.edu.*

Manuscript received 5 Aug. 2003; revised 8 Feb. 2004; accepted 6 June 2004; published online 17 Dec. 2004.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0141-0803.

such as covering the eyes, do not thwart face recognition software. What is needed is an algorithm to de-identify faces in video data such that many facial characteristics remain yet face recognition software cannot reliably identify subjects whose images are captured in the data. This work precisely defines the problem and provides an algorithm (k-Same) that is an effective solution. The algorithm guarantees privacy protection by identifying the "closest" k faces in a video clip and then replacing each of the k faces with the same "averaged" face. By sharing these minimally distorted de-identified images, we can restore the current expectation of privacy so that society does not have to choose sharing video data over privacy, but society can share video data freely with privacy protection.

2 Face De-Identification Definitions

This is the first work to formally introduce the principles of face de-identification. It is imperative therefore to precisely define terms and common expressions from this vantage point.

In face recognition, faces are detected in the raw video image to provide face stills (Definition 2.1) and a "registration" process performed to provide a face image (Definition 2.2). A face still is normalized, rotated, and cropped, as needed, to provide a face image. The goal is to adjust for pose and make the location of eyes and their interpupil distance align in roughly the same position in each image, thereby making face images somewhat comparable to one another. Samples are provided in Example 2.1. The preprocessing needed to get a face image from a face still is one of the greatest current weaknesses of face recognition [5]. The detection of faces, the localization of face stills, and the registration of face stills into face images are all considered preprocessing to this work on de-identification. Therefore, the methods in this work do not exploit current weaknesses in detection and registration.

Definition 2.1 (Face Still). *A **face still** is a column vector* $\mathbf{P}$ *of size* N_{pic}*. Each cell in* $\mathbf{P}$ *stores a value from 0 to 255, inclusive, which is the gray-scale pixel value reporting intensity. The image displayed in a face still includes only one person's face.*

Definition 2.2 (Face Image). *A **face image** (or simply "face" or "image") is a column vector* Γ *of size* N*. Each cell in* Γ *stores a color coding for a pixel. In grayscale, a pixel value is from 0 to 255, inclusive, which is the gray-scale intensity. A face image contains a normalized image of only one person's face. A face image removes much of the hair, clothes, and background from a face still.*

Example 2.1 (Face Still and Face Image). Fig. 1b shows a face image normalized from the face still in Fig. 1a. The face image in this example is a column vector having 13,266 pixels and is displayed graphically as a rectangle having 99 columns and 134 rows.[1]

A face set (Definition 2.3) is a set of M face images that can also be represented as a matrix. It is convenient in privacy discussions to have a single face in a face set relate to one person. This is the idea of a "person-specific" face set (Definition 2.4). Example 2.2 provides a sample.

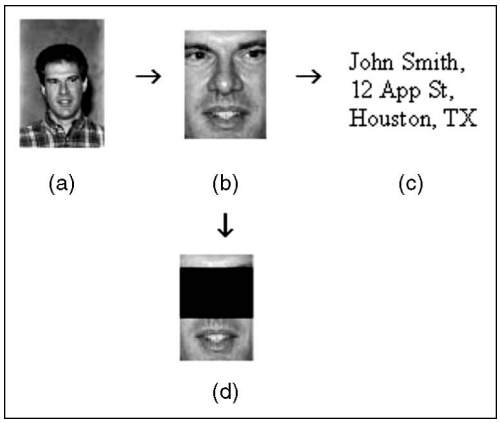


Fig. 1. (a) Face still is (b) normalized to face image, then (c) identified to provide the name and address of the subject, and (d) the face image is de-identified.

Definition 2.3 (Face Set). *A **face set** is a set of* M *face images,* $\{\Gamma_i : |\Gamma_i| = N, i = 1, \ldots, M\}$*. The result is analogous to a matrix of* M *columns and* N *rows. Each column is a face image. Each row corresponds to the same pixel location in each face image. Each element is a pixel value.*

Definition 2.4 (Person-Specific Face Set). *Let* $\mathbf{H}$ *be a face set having* M *images,* $\{\Gamma_1, \ldots, \Gamma_M\}$*.* $\mathbf{H}$ *is **person-specific** if and only if each* $\Gamma \in \mathbf{H}$ *relates to only one person and no two images* $\Gamma_1 \in \mathbf{H}$*,* $\Gamma_2 \in \mathbf{H}$ *relate to the same person.*

Example 2.2 (Person-Specific Face Set). The top of Fig. 2 shows two copies of a person-specific face set, $\mathbf{H}$. There are two face images in $\mathbf{H}$. Each image in $\mathbf{H}$ relates to a distinct person. The two images in $\mathbf{H}$ do not relate to the same person. The images are displayed graphically, but are stored as two column vectors having 13,266 pixels. $\mathbf{H}$ is stored a matrix having 13,266 rows and two columns.

Both a face set and a person-specific face set may have duplicates, the latter being realized by the appearance of identical twins, for example. Therefore, a face set is a multiset, which is a set that maintains duplicates.

Face identification results when a face image is properly associated with explicit identifiers, such as name and address, of the person who is the subject of the face image. Explicit identification is a grave privacy concern. Fig. 1c shows the results of face identification in which Fig. 1b is identified as "John Smith." "Face recognition" as opposed to "face identification," relates a face image to a "known" face image, which may or may not be explicitly identified. Recognizing a face is not necessarily the same as identifying a person because the identities of the subjects of a face set may not necessarily be known. For example, consider a robber's image caught on video during a robbery. The identity of the robber may not be known, even though the robber's image may be recognized as also appearing on a video clip at a subway stop. The faces are recognized as belonging to the same person, even though the identity of the person is not known. Face recognition is the primary concern in this work.

1. What is traditionally done in face recognition to produce a column vector from a rectangular appearing image is to append the values appearing in each row, from left to right. In closed form, the cell locations in the column vector is $(i-1)^*N + j$ for any i,j in the matrix.

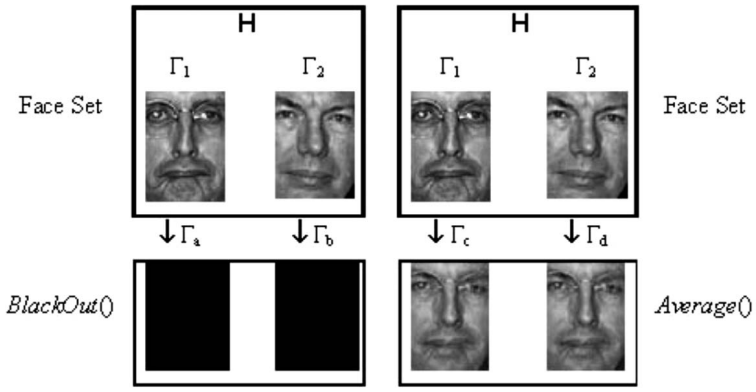


Fig. 2. Face set **H** with effective de-identification using *BlackOut*() and preserved face de-identification using *Average*().

Definition 2.5 (Face Recognition Software). *Given a face set* $\mathbf{H}$ *and a face image* Γ, ***face recognition software*** *is a program that returns the image in* $\mathbf{H}$ *best matching* Γ.

The goal of this work is to alter face images in such a way that face recognition software cannot be reliably performed on the resulting images. Altering face images to conceal identity is termed face de-identification (Definition 2.6). An example of face de-identification appears in Fig. 1d in which the eyes and nose are covered.

Definition 2.6 (Face De-Identification). *Let* $\mathbf{H}$ *and* $\mathbf{H_d}$ *be face sets,* $\Gamma \in \mathbf{H}$, $\Gamma_{\mathrm{d}} \in \mathbf{H_d}$, $f : \mathbf{H} \rightarrow \mathbf{H_d}$ *be a function that attempts to conceal the identity of the subject of the original face image; and,* $f(\Gamma) = \Gamma_{\mathrm{d}}$ *but* $\Gamma \neq \Gamma_{\mathrm{d}}$ *(element-wise).* f *is termed* ***face de-identification*** *("de-identification," "de-identification function").* Γ_{d} *is a de-identified image.*

De-identifying a face image may provide some privacy protection, but the act of de-identification itself provides no privacy assurance. Other details may remain in the image that allow the subject to be reidentified. For example, different ad hoc efforts have attempted to mask identities of people during television interviews. At least one de-identification attempt failed to provide adequate protection and resulted in a lawsuit [24]. Later in this paper, it is shown that many common ad hoc efforts at masking faces do not thwart face recognition software. De-identification alone is not sufficient. Effective de-identification (Definition 2.7) de-identifies faces with a provable privacy assurance. Example 2.3 presents *BlackOut*() as an effective de-identification that provably restricts face recognition.

Definition 2.7 (Effective De-Identification). *Let* $\mathbf{H}$ *be a person-specific face set;* $\mathbf{H_d}$ *be a face set;* $f : \mathbf{H} \rightarrow \mathbf{H_d}$ *be the transformation function used in face de-identification, such that* $f(\Gamma) = \Gamma_{\mathrm{d}}$, *where* $\Gamma \in \mathbf{H}$ *and* $\Gamma_{\mathrm{d}} \in \mathbf{H_d}$; g *be a face identification relation* $g : \mathbf{H_d} \rightarrow \mathbf{H}$; *and,* C *be a provable claim about* f*'s ability to restrict face identification (or face recognition) by* g. *The function* f *provides* ***effective de-identification*** *with respect to* C *and* f *is said to be* ***effective***. *If* f_1 *and* f_2 *are effective with respect to the same* C, *then* f_1 *and* f_2 *are considered* ***equally effective*** *with respect to* C.

Example 2.3 (Effective De-Identification). Let $\mathbf{H}$ be a person-specific face set and *BlackOut*() be the de-identification function defined as:

Given person-specific face set $\mathbf{H}$ and face image $\Gamma \in \mathbf{H}$ having N rows, return $[v_1, \ldots, v_N]$, where each $v_i = 0$.

BlackOut() returns a face image where each of the N cells has 0 (the color black in gray scale). Let f be

$$BlackOut : \mathbf{H} \rightarrow \mathbf{H_d},$$

g be a relation such that $g : \mathbf{H_d} \rightarrow \mathbf{H}$, and C be:

Given face set $\mathbf{H}$, $|\mathbf{H}| > 1$, $f : \mathbf{H} \rightarrow \mathbf{H_d}$, and face image Γ_2, where $\Gamma_2 = f(\Gamma_1)$ for $\Gamma_1 \in \mathbf{H}$, $\Gamma_2 \in \mathbf{H_d}$, Γ_1 cannot be uniquely determined.

The correctness of *BlackOut*()'s claim C is straightforward. Any relation g will relate to all members of $\mathbf{H}$ indistinctly.

Fig. 2 shows a face set $\mathbf{H}$ de-identified by *BlackOut*(). $BlackOut(\mathbf{H}, \Gamma_1) = \Gamma_{\mathrm{d}}$ and $BlackOut(\mathbf{H}, \Gamma_2) = \Gamma_{\mathrm{d}}$. Given a resulting de-identified face set $\{\Gamma_{\mathrm{d}}, \Gamma_{\mathrm{d}}\}$,[2] no human or machine can determine whether the subject of the image is Γ_1 or Γ_2 because Γ_{d} is the same for both. Correct face recognition is limited to guessing with probability $1/|\mathbf{H}|$.

BlackOut() is certainly an effective privacy guard in settings in which no facial details are to be provided. But, some settings, as described in the Introduction, may require effective de-identification that maintains facial details in the image. Let $F = \{f_i\}$ be a set of effective de-identification functions. Of all $f_i \in F$, preference is made for those that maintain the most detail in the image.

A metric named *loss*() captures information loss resulting from de-identification. *loss*() is monotonic and can be based on entropy, percentage of changed pixels, or some other scheme. *loss*() compares two face images, Γ_1 and Γ_2, (or two face sets) and reports the amount of distortion between them. The greater the value of *loss*(), the greater the information loss. Minimum *loss*() is realized when the images are the same. Let f be a de-identification function, and $f(\Gamma_1) = \Gamma_2$, $loss(\Gamma_1, \Gamma_2) \geq loss(\Gamma_1, \Gamma_1)$. The *loss*() metric helps describe "preserved face de-identification" in Definition 2.8. Example 2.4 presents *Average*(), which is as effective as *BlackOut*() but more preservative because its results have less information loss.

2. Recall that face sets are multisets in which duplicates are maintained.

Definition 2.8 (Preserved Face De-Identification). *Let* $\mathbf{H}$ *be a person-specific face set;* $\mathbf{H_d}$ *be a face set;* $F = \{f_i\}$ *be a set of equally effective de-identification functions where each* $f_i : \mathbf{H} \rightarrow \mathbf{H_d}$*;* $\Gamma \in \mathbf{H}$*; and, loss() be a precision metric related to* F*. If for any* $f_i \in F$*, there does not exist* $f_j \in F$ *such that* $loss(\Gamma, f_j(\Gamma)) < loss(\Gamma, f_i(\Gamma))$*, then* f_i *is a* ***preserved face de-identification*** *(or "preserved de-identification" over* F *with respect to loss(). It is said that* f_i *is "preservative." If* $loss(\Gamma, f_i(\Gamma)) = loss(\Gamma, f_j(\Gamma))$*, then* f_i *and* f_j *are* ***equally preservative*** *with respect to loss() and* F*.*

Example 2.4 (Preserved Face De-Identification). Let $\mathbf{H}$ be a person-specific face set and *Average*() be the de-identification function defined as:

Given person-specific face set $\mathbf{H}$ and face image $\Gamma \in \mathbf{H}$ having N rows, return:

$$\Psi = \left[\frac{\sum_{j=1}^{|\mathbf{H}|} \Gamma_j[1]}{|\mathbf{H}|}, \ldots, \frac{\sum_{j=1}^{|\mathbf{H}|} \Gamma_j[N]}{|\mathbf{H}|} \right] = \frac{1}{|\mathbf{H}|} \bullet \sum_{j=1}^{N} \Gamma_j. \quad (1)$$

Average() returns a face image (Ψ), where each pixel in Ψ is the sum of the value for that pixel in all the faces of $\mathbf{H}$ divided by the number of faces in $\mathbf{H}$. $Average : \mathbf{H} \rightarrow \mathbf{H_d}$ is effective with respect to claim C defined in Example 2.3. Any relation $g : \mathbf{H_d} \rightarrow \mathbf{H}$ relates to all members of $\mathbf{H}$ indistinctly because each face is replaced with the same average face Ψ. Fig. 2 shows a face set $\mathbf{H}$ de-identified by *Average*(). $Average(\mathbf{H}, \Gamma_1) = \Gamma_c$ and $BlackOut(\mathbf{H}, \Gamma_2) = \Gamma_d$. Given a resulting face image $\Gamma \in \{\Gamma_c, \Gamma_d\}$, no human or machine can determine whether the subject of the image is Γ_1 or Γ_2 because both Γ_c and Γ_d are Ψ. As with *BlackOut*(), correct face recognition is limited to guessing with probability $1/|\mathbf{H}|$. *BlackOut*() and *Average*() are equally effective. But, *Average*() retains more facial details than *BlackOut*().

Let Γ_1 and Γ_2 be face images of size N. Euclidean distance, *euclid*(), is a loss metric defined as the square root of the sum of the square of the differences in pixel values in the faces:

$$euclid(\Gamma_1, \Gamma_2) = \sqrt{\sum_{i=1}^{N} \left| \Gamma_1[i] - \Gamma_2[i] \right|^2}. \quad (2)$$

Fig. 2 shows results from *BlackOut*() and *Average*() on $\mathbf{H}$. For $BlackOut()$, $euclid(\Gamma_1, \Gamma_a) = 14,187$ and $euclid(\Gamma_2, \Gamma_b) = 14,869$. For $Average()$, $\Psi = \Gamma_c = \Gamma_d$. $euclid(\Gamma_1, \Psi) = 2,143$ and $euclid(\Gamma_2, \Psi) = 2,153$. *Average*() is preservative in comparison to *BlackOut*().

A de-identification function can provide images that retain a lot of detail, but such functions may not be effective and therefore are not preservative. For example, a de-identification function that de-identifies by changing a randomly selected nonblack pixel to black, maintains many facial details, but it does not respect claim C in Example 2.3, so it is not as effective as *BlackOut*() and *Average*() and, therefore, cannot be preservative with respect to them.

The effectiveness of *BlackOut*() and *Average*() relies on resulting images relating ambiguously to all members of the original person-specific face set ($\mathbf{H}$). A substantive improvement involves having each de-identified image relate ambiguously to k members of the original face set, where $2 <= k <= |\mathbf{H}|$. This provides *k*-anonymity privacy protection, which was introduced in earlier work on field-structured data [21]. De-identifying each face in $\mathbf{H}$, maintaining duplicate faces, provides a face set $\mathbf{H_d}$ that adheres to *k*-anonymity if and only if each face image appearing in $\mathbf{H_d}$ appears at least k times; see Definition 2.9. Samples of *k*-anonymized face sets are provided in Examples 2.5 and 2.6.

Definition 2.9 (*k*-anonymity on Face Images). *Given a person-specific face set* $\mathbf{H}$*, a set of de-identified face images* $\mathbf{H_d}$ *in which duplicates are maintained (*$\mathbf{H_d}$ *is a "multiset"),* $|\mathbf{H}| > 1$*,* $|\mathbf{H}| = |\mathbf{H_d}|$*, a de-identification function* $f : \mathbf{H} \rightarrow \mathbf{H_d}$*, and* $g : \mathbf{H_d} \rightarrow \mathbf{H}$ *a relation that is the inverse of* f*. If for each* $\Gamma \in \mathbf{H}$ *there exists* $\Gamma_d \in \mathbf{H_d}$*, where* $f(\Gamma) = \Gamma_d$ *and for each* $\Gamma_d \in \mathbf{H_d}$*,* $|g(\Gamma_d) = \Gamma| \geq k$*, then* $\mathbf{H_d}$ *adheres to* ***k-anonymity****. It is said that* $\mathbf{H_d}$ *is* ***k-anonymized*** *over* $\mathbf{H}$*.*

Example 2.5 (*k*-anonymity on Face Images). Let $\mathbf{H} = \{\Gamma_1, \ldots, \Gamma_{100}\}$ be a person-specific face set of 100 images where $|\Gamma_i| = N$ for $i = 1, \ldots, 100$ and let $k = 2$. *Average*() is applied to each face in $\mathbf{H}$ such that $Average(\mathbf{H}, \Gamma_j) = \Psi_{j,j+1}$ and $Average(\mathbf{H}, \Gamma_{j+1}) = \Psi_{j,j+1}$ for $j = 1, 3, \ldots, 99$. $\Psi_{j,j+1}$ is the average face computed for Γ_j and Γ_{j+1}. The resulting de-identified face set is

$$\mathbf{H_d} = \{\Psi_{1,2}, \Psi_{1,2}, \ldots, \Psi_{99,100}, \Psi_{99,100}\}.$$

$|\mathbf{H_d}| = 100$ images. $\mathbf{H_d}$ is *k*-anonymized, where k is 2.

Example 2.6 (*k*-anonymity on Face Images). Let $\mathbf{H} = \{\Gamma_1, \ldots, \Gamma_{100}\}$ be a person-specific face set of 100 images where $|\Gamma_i| = N$ for $i = 1, \ldots, 100$. *BlackOut*() is applied to each face in $\mathbf{H}$ such that $BlackOut(\mathbf{H}, \Gamma_j) = [0_1, \ldots, 0_N]$ for $j = 1, \ldots, 100$. The resulting de-identified face set $\mathbf{H_d}$ has 100 face images in which all pixel values are 0 (black). $\mathbf{H_d}$ is *k*-anonymized, where k is $2, 3, \ldots, 100$.

Partitioning a person-specific face set into smaller face sets or "clusters" of at least k faces provides privacy protection because an aggregate face is published for the members of each cluster in lieu of their original images. Basing each aggregate face on a cluster of homogeneous original faces minimizes information loss. One concern is finding optimal clusters—those having the k most homogeneous faces. A distance measure is used to determine closeness.

A face image has been represented as a column vector having N cells, but it can also be represented as a point in N-dimensional space by viewing the vector as a tuple. A measure, such as Euclidean distance (2), can be used to compute distances between points to determine clusters of closest neighboring points (or faces).

Once minimally distant clusters are found, aggregate face images must be constructed in such a way as to minimize information loss. Many strategies are possible. This work constructs an aggregate face as the N-dimensional (or pixel-wise) "average" of the cluster. In data mining, this is termed a geometric centroid (or center of mass) [25]. Equation (1) computes Ψ as the centroid of cluster $\mathbf{H}$.

In summary, this work seeks preserved de-identification that thwarts face recognition software by enforcing k-anonymity. This is termed the k-Same problem (Definition 2.10).

Definition 2.10 (k-Same). *Given a person-specific face set* $\mathbf{H}$*; and, a face set* $\mathbf{H_d}$ *which is k-anonymized over* $\mathbf{H}$ *using a preserved face de-identification function* $f : \mathbf{H} \rightarrow \mathbf{H_d}$*, if* f *is effective with respect to the claim:*

Given any face image $\Gamma_d \in \mathbf{H_d}$, where $\Gamma_d = f(\Gamma)$ for $\Gamma \in \mathbf{H}$, there cannot exist any face recognition software for which the subject of Γ_d can be correctly recognized as Γ with better than $1/k$ probability.

Then, f *is a* ***k-Same de-identification function*** *and* $\mathbf{H_d}$ *is a* ***k-Same de-identification****. The goal is to determine the appropriate function f with minimal information loss.*

Partitioning a person-specific face set into k-sized clusters of faces achieves k-anonymity. Each cluster is replaced by a single image that will be replicated to represent each image in its cluster. If the faces in a cluster are averaged together to produce the replacement image, then the averaged face may retain characteristics more similar to one image than another in the cluster. However, because all images in the cluster are replaced by the same averaged face, correctly relating the k occurrences of the averaged face can at best be correct 1 in k times. The recognition would be incorrect for the other replacement images. This is the approach for achieving k-Same taken in this paper. It reduces to being the same problem as microaggregation in statistical disclosure control. Recent work [13] on microaggregation has shown that the optimal selection of k-sized clusters of minimally distant N-dimensional points, where $N > 1$, is an NP-hard problem. A corollary is that no algorithm can run in reasonable time to find optimal groups of "closest" faces for k-Same clusters.

The problem addressed in this work is not to determine the optimal size of clustering or how to divide faces optimally into k clusters. These are both classical problems in statistics and computer science. The problem in this work is to find the optimal selection of faces to form as many clusters as needed provided each cluster has k closest faces.

In the next section, the method and operating paradigm of face recognition software is introduced. Then, two heuristic solutions, k-Same-Pixel and k-Same-Eigen, are presented as ways to thwart face recognition software. Following that are experimental results. This paper ends with a discussion on related work and a discussion on the general applicability of this work.

3 Face Recognition Software

The current baseline face recognition algorithm is "Eigenfaces," also known as Principal Components Analysis (PCA) [23]. Eigenfaces is the technique against which others are measured and is the basis of methods explored in this paper. A description of how Eigenfaces work is provided, but first, some basic terms are described.

Three face sets are traditionally used in face recognition software [11]. **Training** is a face set used to initially learn parameters about the faces to classify. **Gallery** is a face set of known faces. **Training** and **gallery** need not be the same. **Probe** is a face set of faces to "recognize" by selecting best matches in **gallery**. Subjects in **probe** are not necessarily also in **gallery**.

Recognition of a face in **probe** is a rank ordering of the faces in **gallery** where the "closest" face appears first and the least similar face appears last. The **performance** of a face recognition program is defined as the percentage of faces in **probe** correctly matching their e closest **gallery** images [23], where $e \geq 1$. When e is 1, only the single "best matched" results.

Supporting methods for *Eigenfaces*() appears in Fig. 3. Here is an overview before examining the details. Initialization consists of mapping all images in **gallery** into a face space characterized by the faces in **training**. This is done by executing: *Setup*(**gallery**, **training**) which generates the face space. Then, for each face in **probe**, display the image in **gallery** that best matches it in that face space. Having a loop that iterates through each face in $\mathbf{probe}(\Gamma \in \mathbf{probe})$ does this. The body of the loop has two steps: 1) $\mathrm{match} = Recognize(\Gamma)$ and 2) $\mathrm{print}\ \Gamma, \mathrm{match}[1][1]$, to display the face and its best match.

In the next paragraphs, details of these methods are explained. Throughout this discussion, each face image in **probe**, **gallery**, and **training** has N pixels and the number of face images in training is M.

The goal of *Setup*() is to create an Eigen "face space" based on the faces in **training** and then, projects images from **gallery** into the face space. An "average face," Ψ, is computed across the M face images in **training** (Fig. 3a, line 2). The "average face," Ψ, is subsequently taken away from each image, Γ_i in **training**, yielding difference faces, Φ_i (Fig. 3a, lines 3-4). The set of these "difference" faces is an N by M matrix, $\mathbf{A}$ (Fig. 3a, line 5).

A covariance (or scatter) matrix is the product of the transpose of $\mathbf{A}$ with itself $\mathbf{AA^T}$, but due to the structure of face recognition data, namely, $M << N$, the variable $\mathbf{C}$ is computed using $\mathbf{A^TA}$, yielding a square matrix of dimension M [20]. The eigenvectors (or "eigenfaces") of $\mathbf{C}$ are determined by satisfying the characteristic equation, where the determinant of the matrix less the eigenvalues along the diagonal elements is equal to zero using the method of Lagrange multipliers. See Fig. 3a, lines 7-8. For more in-depth details of lines 7 and 8, refer to [23].

"Eigenfaces" is the resultant multidimensional face-space characterized by the Eigenvectors and the average face. All images in **gallery** are projected into the face space and their original and projected images saved in a matrix, **facespace** (Fig. 3a, lines 9-14). Fig. 3b, line 1 shows the expression for projecting an image into eigenfaces.

The result of *Setup*() is **facespace**, the average face (Ψ), and the eigenvectors $\mathbf{V}$. These are used by *Recognize*(), in Fig. 3c, to match faces as follows: Face images in **probe** are projected through the eigenfaces (Fig. 3c, line 1). The distance between the projected probe image and each projected gallery image is computed (typically using Euclidean distance or Mahalanobis distance). A face, $\Gamma \in \mathbf{probe}$, is recognized as the face in **gallery** whose projected image is closest to Γ's projected image. *Recognize*() returns all the images in **gallery** sorted by closeness of projected images to Γ's projected image (Fig. 3c, line 3). This is a copy of **facespace** sorted in rank order of best matches.

Algorithm: *Setup*(gallery, training) *// initialization*

Input: Face sets **gallery** and **training**.
Output: facespace, a 2-dimensional matrix. Each row has a pair of face images. Column [1] is a face from **gallery** and column [2] is that face projected in "face space". Each face in **gallery** is present once in the matrix, so there are |**gallery**| rows.

Steps

let M = |**training**|		1
Ψ = *Average*(**training**)	*// defined in Equation 1*	2
for *each* $\Gamma_i \in$ **training**		3
$\Phi_i = \Gamma_i - \Psi$	*// difference of average face*	4
let A = $[\Phi_1, \ldots, \Phi_M]$	*// matrix of difference vectors*	5
let C = **A** $\mathbf{A}^T$	*// variation of covariance matrix*	6
L[i]= λ_i such that $\|\mathbf{C}-\lambda_i\mathbf{I}\|=0$ for i = 1,…, *N*	*// eigen values*	7
V[i]= $\mathbf{v}_i$ such that $\mathbf{C}\mathbf{v}_i = \lambda_i\mathbf{v}_i$ for i = 1,…, *N*	*// eigen vectors*	8
for i=1 to |**gallery**| **do:**		9
let $\Gamma \in$ **gallery**	*// select a face from gallery*	10
gallery = **gallery** - {Γ}		11
facespace[i][1]= Γ	*// store face*	12
facespace[i][2]=*Project*(Γ)	*// projected face*	13
return facespace		14

(a)

Algorithm: *Project*(Γ) *// projecting a face image into Eigenfaces' face space*

Input: Face image Γ
Output: a face image, which is Γ in the "face space" characterized by the previously derived eigenvectors and average face from the training set.
Uses: Eigenvectors **V** and average face Ψ
Steps

Ω[i] = $\mathbf{V}[i]^T(\Gamma-\Psi)$ for i = 1,…, *M* *// store weights using only top eigenvectors*	1
return Ω	2

(b)

Algorithm: *Recognize*(Γ) *// recognizing a face image*

Input: Face image Γ
Output: a copy of **facespace** where the rows are sorted in order of closeness to Γ's projected face. The most similar face appears in the first row and the least similar face appears in the last row.
Uses: **facespace** and a distance function *dist*()

Steps

O = *Project*(Γ)	*// project into face space*	1
let match = **facespace**	*// make a copy*	2
Sort **match** by row based on *dist*(O, **match**[][2])	*// sort projected images*	3
return match		4

(c)

Fig. 3. Supporting methods for Eigenfaces.

4 *k*-Same-Pixel and *k*-Same-Eigen

This section presents *k-Same-Pixel*() and *k-Same-Eigen*(), which are effective at limiting the face recognition software's ability to reliably recognize faces, even when Eigenfaces is not the face recognition software used. They are not optimal *k*-Same solutions because there may exist other equally effective de-identification functions that maintain more detail in the resulting images.

4.1 *k-Same-Pixel*() and *k-Same-Eigen*()

Fig. 4 presents the *k-Same-Pixel*() algorithm. Given an original person-specific face set, **H**, and a value k, the algorithm returns a *k*-Same face set over **H** with respect to k. Using Eigenfaces' *Setup*() routine, **H** is used to generate an overall average face and the projected gallery images against which faces will be recognized (line 1).

Each face in **H** is de-identified in lines 4 through 10 as follows: For a given face, *Recognize*() provides a matrix containing all the face images, not yet de-identified, sorted by closeness to the given face. The top k rows of the matrix, representing the k closest faces, are averaged together in line 7. k copies of this average face are added to the solution set, $\mathbf{H}_d$, in line 8. The k faces are then removed from **H** (line 9) and from **facespace** (line 10) because they have all been de-identified. On each iteration of the loop, the number of faces to de-identify is reduced by k. In case the number of original face images in **H** is not evenly divisible by k, an adjustment to k is made in line 5 on the last iteration, to group up to $2k-1$ faces together. When

Algorithm: *k-Same-Pixel*(H, k)

Input: Person-specific face set named **H** and a k privacy constraint
Output: a k-Same face set named $\mathbf{H_d}$
Assumes: $|\mathbf{H}| \geq k$
Uses: Eigenfaces pseudo-code and its **facespace** variable

Steps	
Setup(**H**, **H**) // *H is gallery and training*	1
let $\mathbf{H_d}$ be an empty multi-set	2
for $\Gamma \in$ **H do:**	3
match = *Recognize*(Γ)	4
if $|\mathbf{H}| < 2k$ **then** $k = |\mathbf{H}|$ // *group all remaining together*	5
let closest be the face set containing **match**[1][1],…,**match**[k][1]	6
Ψ = *Average*(**closest**)	7
Add k copies of Ψ to $\mathbf{H_d}$	8
Remove faces **match**[1][1], …, **match**[k][1] from **H**	9
Remove rows **match**[1][], …, **match**[k][] from **facespace**	10
return $\mathbf{H_d}$	11

Fig. 4. De-identification algorithm *k*-Same-Pixel averages k closest gallery images.

execution concludes, the resulting face set, $\mathbf{H_d}$, has at least k occurrences of each face image contained within it. Each image in $\mathbf{H_d}$ is the average of k closest faces.

In *k-Same-Pixel*(), the de-identified image is based on the pixel-wise average of the original face images. A variation is to use the projected images as the basis for averaging. This is done in *k-Same-Eigen*(). Line 6 is replaced in Fig. 4 with: **let closest** be the face set containing $\mathbf{match}[1][2], \ldots, \mathbf{match}[k][2]$. That is the only difference between *k-Same-Pixel*() and *k-Same-Eigen*(). Images from *k-Same-Eigen*() have a blurred effect compared to those from *k-Same-Pixel*() because the projected images used in *k-Same-Eigen*() retain only the most important characteristics. Less important components have been removed.

Fig. 7a shows face images de-identified by *k-Same-Pixel*() and *k-Same-Eigen*() over the same subjects for $k = 2, 3, 5, 10, 50$, and 100. The number of faces in the original face set is 200.

4.2 Correctness of *k-Same-Pixel*() and *k-Same-Eigen*()

Let $\mathbf{H}$ be a person-specific face set, $|\mathbf{H}| > 1$, k be a privacy constraint, $k > 1$, $|\mathbf{H}| \geq k$, and $\mathbf{H}_d$ be the result from $k\text{-}Same\text{-}Pixel(\mathbf{H}, k)$, or alternatively from

$$k\text{-}Same\text{-}Eigen(\mathbf{H}, k).$$

Theorem 1 states that $\mathbf{H_d}$ is k-anonymized over $\mathbf{H}$. Theorem 2 states that *k-Same-Pixel*() is effective at thwarting face recognition software. There may exist some other *k*-same de-identification function, other than *k-Same-Pixel*(), however, that can have less information loss.

Theorem 1. *If* $\mathbf{H}$ *is a person-specific face set,* $|\mathbf{H}| > 1$, k *is a privacy constraint,* $k > 1$, $|\mathbf{H}| \geq k$, *and* $\mathbf{H_d} = k\text{-}Same\text{-}Pixel(\mathbf{H}, k)$, *then* $\mathbf{H_d}$ *is k-anonymized over* $\mathbf{H}$.

Proof. Fig. 4 contains pseudocode for *k-Same-Pixel*(). In each iteration of the loop contained in lines 3 through 10, k faces are added to $\mathbf{H_d}$ (see line 8), and k faces are removed from $\mathbf{H}$ (see line 9). On the last iteration, k is adjusted from its original value, k_0, if needed, to be the number of faces remaining (line 5) in $\mathbf{H}$. On the last iteration, $k_0 \leq k < 2k_0$. When the loop ends, the number of faces originally in $\mathbf{H}$ is the number of faces in $\mathbf{H_d}$. On each iteration, k copies of the same averaged face (Ψ in line 7) are added to $\mathbf{H_d}$ (line 8), so the k copies are indistinguishable. The k copies of the average face (Ψ) have a one-to-one correspondence to the original k face images in $\mathbf{H}$ (see lines 6 and 7) that composed Ψ. In summary, 1) $|\mathbf{H}| = |\mathbf{H_d}|$, 2) for each $\Gamma \in \mathbf{H}$ there exists $\Psi \in \mathbf{H_d}$, and 3) for each $\Psi \in \mathbf{H_d}$, there are k original faces on which it was based. □

Theorem 2. *If* $\mathbf{H}$ *is a person-specific face set,* $|\mathbf{H}| > 1$, k *is a privacy constraint,* $k > 1$, $|\mathbf{H}| \geq k$, $\mathbf{H_d} = k\text{-}Same\text{-}Pixel(\mathbf{H}, k)$, *and* $\Gamma_d \in \mathbf{H_d}$, *there cannot exist any face recognition software for which the subject of* Γ_d *can be correctly recognized in* $\mathbf{H}$ *with better than* $1/k$ *probability.*

Proof. The proof is straightforward, but is expounded to demonstrate characteristics of the privacy protection provided. Let $g : \mathbf{H_d} \rightarrow \mathbf{H}$ be a software recognition program, and $\Psi_d \in \mathbf{H_d}$. Let $\mathbf{P}$ be a face set containing the subjects of Ψ_d; that is, $\mathbf{P} = \{\Gamma_i | \Gamma_i \in \mathbf{H}$ and $k\text{-}Same\text{-}Pixel(\Gamma_i) = \Psi_d$, for $i = 1, \ldots, k\}$. There are three cases to consider.

Case 1 (k best choices are all onto $\mathbf{P}$). Each $g(\Psi_d)$ is a vector whose best matching k faces are the same (and only the same) as the faces in $\mathbf{P}$. The k faces in $\mathbf{P}$ are the subjects of the k occurrences of Ψ_d. The Ψ_ds are indistinguishable, so correctly assigning ("recognizing") a Ψ_d to the face in $\mathbf{P}$ that was its subject has probability $1/k$.

Case 2 (k best choices are not all onto $\mathbf{P}$). Let $\mathbf{A}$ be the face set containing the k best matches of $g(\Psi_d)$. $\mathbf{A}$ includes some faces in $\mathbf{P}$ and some not in $\mathbf{P}$. Assigning ("recognizing") a Ψ_d to the face in $\mathbf{A}$ that was its subject is probability 0 if the subject is not present in $\mathbf{A}$ and $1/k$ otherwise.

Case 3 (matching each face in $\mathbf{P}$). If $\mathbf{P}$ is known and $g(\Psi_d)$ claims its best match to be Γ, $\Gamma \in \mathbf{P}$, then for each of the k faces $\Gamma_i \in \mathbf{P}$, declaring Ψ_d to be recognized as each Γ_i will only be correct once, so the probability of a correct recognition is $1/k$. If $\Gamma \notin \mathbf{P}$, then the probability of a correct recognition is 0.

In all cases, the probability of correctly recognizing a face was no better than $1/k$. □

The basis of the validity of Theorem 2 is realized because k faces are replaced with the same replacement image. The recognition task is then left to disambiguate the k occurrences of the same image to k or more people.

Theorem 2 shows that, even though methods of *Eigenfaces*() is used by *k-Same-Pixel*() and *k-Same-Eigen*(), their ability to provide *k*-anonymized solutions is invariant to the face recognition software that attempts to recognize faces from their results. The de-identified faces are not just protected against matrix decomposition-based face recognition technology. *k-Same-Pixel*() and *k-Same-Eigen*() can protect against any face recognition system, including technologies that have yet to be designed. Let $\mathbf{H}$ be a person-specific face set that is the subject of de-identification. A *k*-anonymized solution over $\mathbf{H}$ reduces the number of distinct faces to $|\mathbf{H}|/k$ prior to a recognition attempt. It is impossible to distinguish between at least k faces in $\mathbf{H}$ for any face in the *k*-anonymized solution over $\mathbf{H}$.

Optimal clusters are not necessarily found in *k-Same-Pixel*() or *k-Same-Eigen*(). If each face in a cluster, selected the same, and only the same, set of k faces as its closest faces, then *k-Same-Pixel*() would provide an optimal solution. But, this partitioning is a special and rare case. More often clusters overlap, making the best choice of minimally distant clusters difficult to determine. *k-Same-Pixel*() does not attempt to disambiguate these nested cluster preferences. Instead, one face is randomly selected and its k closest faces are grouped together. A different selection of faces typically reveals different clustering choices over the same face set. For these reasons, *k-Same-Pixel*() can provide results with more information loss than is minimal, and *k-Same-Eigen*() can have worse recognition results because of its additional loss of facial details.

k-Same-Pixel() and *k-Same-Eigen*() do not always make optimal cluster selections, but they do run quickly, in linear time. Even though the composition of *k-Same-Pixel*() and related methods focused on clarity, not operational efficiency, they still demonstrate linear time execution. For details of this complexity claim, see [12].

5 Experiments

5.1 Materials

Experiments were run on images from the U.S. Army's Face Recognition Technology (FERET) database, which is publicly available [15]. The database has 14,126 face stills of 1,199 people. Coordinates for the center of the eyes and tip of the nose are provided for each face still. Face stills were cropped, rotated, and scaled to provide face images using this information. Each resulting face image has 13,266 pixels, displayed graphically as 99 columns and 134 rows.

Numerous editions of Eigenfaces are publicly available. This work used MatLab to run experiments, with "calcpca.m" version 2.4 written by Matthews and edited by Gross.

5.2 Test Design

In all experiments, training, gallery, and probe face sets are person-specific. Each subject appearing in gallery also appears in probe, and vice versa. Unless otherwise noted, 200 face images were randomly chosen for each experiment to make a person-specific face set $\mathbf{H}$ for that experiment. Let $\mathbf{H}_\mathbf{d}$ be the de-identified face set of $\mathbf{H}$. The probability of correctly recognizing a face image in $\mathbf{H}_\mathbf{d}$ to a face image in $\mathbf{H}$ by guessing is $1/|\mathbf{H}| = 0.005$.

Current face recognition algorithms have numerous weaknesses, such as: detecting faces, producing face images from face stills, and recognizing the same person after a lapse in time. These are all issues independent of de-identification. It is crucial to test de-identification in such a way as to not exploit these weaknesses in face recognition software. Therefore, all experiments reported herein test recognition using only a face set of well-formed face images against itself. Because Eigenfaces always yields 100 percent recognition for this setup [14], these experiments show how performance degrades due to de-identification. Recognition performance using Eigenfaces is reported as the percentage of correct recognitions having a single best match.

To test de-identification techniques, consider the options available to an attacker, who attempts to reidentify (by face recognition software) a de-identified face set. There are three different kinds of attacks, corresponding to three different arrangements of recognizing gallery images to probe images, available. These involve matching: 1) original images to altered images, 2) altered images to original images, and 3) altered images to altered images.

The first type of attack, in which original images are matched to altered images, is termed **naïve recognition** in the context of these experiments. No action is taken by the attacker to account for the de-identification affect. The de-identified images are just run through the face recognition software. For experimental soundness herein, the gallery in naïve recognition tests, will be only the original face set of the images subject to de-identification. In the general case, outside of this experimentation, the gallery would presumably include faces beyond those de-identified thereby possibly providing even better de-identification results.

The second type of attack, in which altered images (as gallery) are matched to original images (as probe), is termed **reverse recognition** in the context of these experiments. Reverse recognition assumes the attacker already has a face set containing the original images that were the subjects of de-identification. The goal is to determine a correct one-to-one correspondence. By using the altered images as the gallery, the alterations due to de-identification may decompose and become dispersed through some number of principal components, thereby limiting the affects of the alterations when matching faces.

The third type of attack, in which altered images are matched to altered images, is termed **parrot recognition** in the context of these experiments. Many de-identification techniques can be replicated. For example, placing a black band over the eyes can easily be duplicated. The attacker can invoke the same de-identification technique on his face sets, thereby de-identifying them. The training set and gallery of the attacker are then de-identified also, and recognition matches a de-identified gallery to a de-identified probe. For experimental soundness herein, the gallery in parrot recognition tests, will contain only those images appearing in the probe. In the general case, outside of this experimentation, the gallery would presumably include faces beyond those de-identified in the probe, thereby possibly providing even better de-identification results.

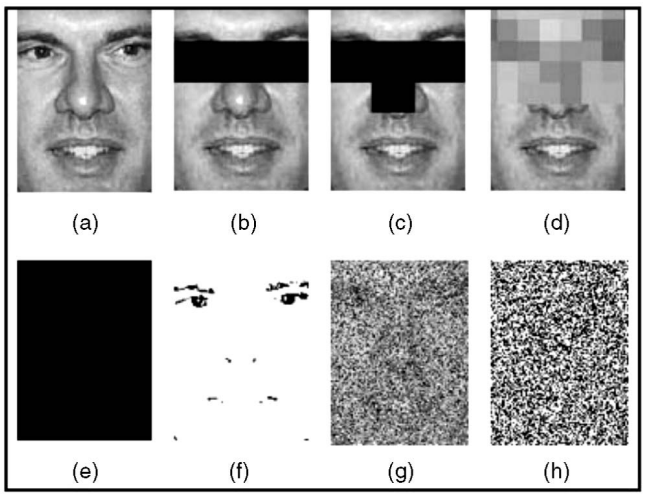


Fig. 5. Examples of (a) an original face image and each ad hoc de-identification method: (b) bar mask, (c) T mask, (d) pixelation, (e) blackout, (f) threshold, (g) random in gray scale, and (h) random in black and white.

5.3 Ad Hoc De-Identification Methods do not Thwart Face Recognition Software

There exist a number of ad hoc de-identification techniques that attempt to mask the identity of a subject in a face image. To the human eye, a masked face image may look sufficiently de-identified. This section reports how several ad hoc de-identification techniques affect the recognition abilities of face recognition software. See [12] for a more comprehensive list of ad hoc methods. Other than blocking out the entire image, *BlackOut*(), these experiments show that ad hoc attempts do not thwart face recognition software.

Face de-identification is not the same as a lack of recognition. Previous experiments have shown that some face recognition algorithms can be robust to different types of image degradation and occlusion, e.g., [7], [10]. However, recognizing degraded or obscured images is not the same as provably de-identifying images. None of the previous experiments attempted to prevent de-identification explicitly, rather they were attempting to identify the current limits of face recognition. In order to test de-identification attempts directly, below are descriptions of the ad hoc de-identification techniques tested and following that are experimental results.

Mask. *BlackOut*: colors the entire face image with either a single color or pattern (see Fig. 5e).

Bar masks: covers the eyes with a single-colored band (see Fig. 5b). *T masks*: cover the eyes and nose with a single color (see Fig. 5c).

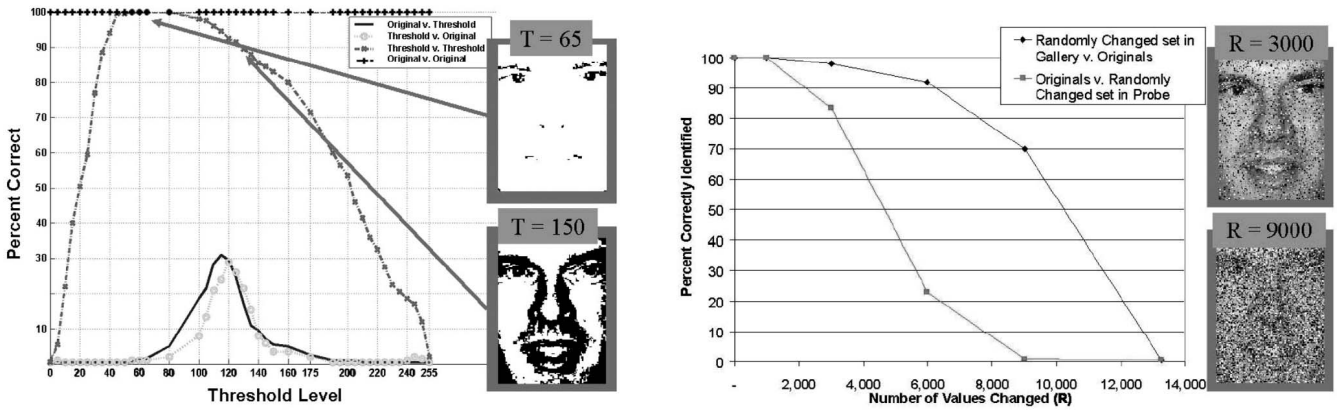
Pixelation. Reduces the number of distinct pixel values in a face image by replacing a square block of pixel values with their averaged value (see Fig. 5d). Experiments were conducted on square pixel blocks of 15, 20, and 30 pixels across.

Random Noise. In black/white images, choose a random pixel position to flip (see Fig. 5h). In gray-scale images, a random value between 0 and 255 replaces a set of randomly chosen pixel positions (see Fig. 5g). The same set of pixels is randomly perturbed in each image.

Threshold. Transforms all gray-scale pixels to either a black or white value (0 or 255, respectively) depending upon a threshold value chosen in the range of 0 to 255 (see Fig. 5f). In additional experiments based on black/white techniques, a threshold value of 65 achieved 100 percent recognition; see Fig. 6a.

Many ad hoc de-identification methods are capable of defeating naïve recognition, which matches original (gallery) to altered (probe) images. The percentage of correct best match recognitions for: 1) bar mask was 2 percent, 2) T mask was 1 percent, and 3) black out was 0 percent. Recall, 100 percent correct recognition was realized when matching original-to-original images. Masking deteriorates this performance. However, using parrot recognition, as described below, bar mask and T mask images have 100 percent correct recognition rates rendering them ineffective de-identification techniques.

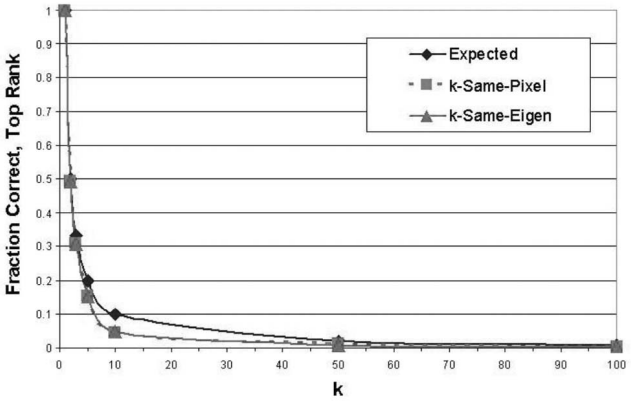
Pixelation is an exception: 99 percent correct recognition was realized when matching original to pixelated images. Despite looking somewhat de-identified to humans (Fig. 5d), and despite pixelation's common use on television to hide faces during interviews, pixelation has virtually no effect on thwarting naïve face recognition software. Pixelation is not



(a) (b)

Fig. 6. Eigenfaces recognition based on best matches using: (a) threshold images and (b) images perturbed by adding random noise.

(a)



(b)

Fig. 7. Face images from (a) *k*-Same-Pixel (top) and *k*-Same-Eigen (bottom) for $k = 2, 3, 5, 10, 50$, and 100 from left to right; (b) performance of naive recognition on faces de-identified by *k*-Same-Pixel and *k*-Same-Eigen.

the only exception. Fig. 6a shows a maximum of 30 percent correct recognition when matching original to threshold images with a threshold of 118.

Fig. 6b shows the results of naïve recognition on images de-identified by adding random noise. The bottom curve shows the recognition rate when matching original to noisy images. When 9,000 of 13,266 (or 68 percent) of the pixels were made noisy, the recognition plummeted to 1 percent, but at 6,000 pixels perturbed, correct recognition was 23 percent. Random noise in black/white images decreases recognition after approximately half of the pixel values were perturbed.

Reverse recognition, which matches altered (gallery) to original (probe) images, performs the same as naïve recognition on most of the ad hoc de-identification techniques tested. There are exceptions. In threshold de-identification, Fig. 6a shows a maximum of 30 percent correct recognition at a threshold of 118 using naïve recognition. Reverse recognition shifts the curve to the right and slightly down. A maximum of 29 percent correct recognition resulted at a threshold of 120 using reverse recognition. More noticeable, differences appear with additive random noise. The bottom curve in Fig. 6b shows the performance of naïve recognition, and the top curve shows the performance of reverse recognition. With 68 percent of the pixels made noisy, naïve recognition was only 1 percent, but reverse recognition was 70 percent.

In parrot recognition, Eigenfaces was retrained on face images having the same alteration and the gallery was also de-identified. Parrot recognition circumvents most of the ad hoc de-identification techniques tested. More than 99.8 percent correct recognition was realized when matching altered-to-altered pixilation images, and 100 percent correct recognition was realized for single bar and T bar images. However, as will be shown in the next section, retraining on *k*-same images will not succeed.

There are exceptions. BlackOut, which provides no information, retains results no better than guessing with parrot recognition. For threshold and additive random noise, results for various levels are shown in Figs. 7a and 7b (for gray-scale), respectively. In Fig. 6a, the threshold experiments of altered to altered (the highest curve in Fig. 6a marked with xs) show that parrot recognition remains above 10 percent within the threshold values of 5 and 250, which are the extremities of the gray-scale.

In summary, the percentage of correct best match recognitions for: 1) bar mask was 2 percent using naïve recognition but 100 percent using parrot recognition, 2) T mask was 1 percent using naïve recognition but 100 percent using parrot recognition, 3) black out was 0 percent using both naïve recognition and parrot recognition, and 4) pixelation was 99.8 percent using naïve recognition and 100 percent using parrot recognition. Therefore,only BlackOut and additive random noise (at approximately 13,000 pixels changed) are equally *effective* with respect to providing no better than 0.5 percent (random guessing) correct recognition. Of these, additive random noise is *preservative* with respect to the Euclidean distance from the original image.

These experiments demonstrate that many ad hoc de-identification techniques may look convincing to human eyes, but in general, they provide little or no protection from face recognition software. With the exception of BlackOut, significant numbers of images were correctly recognized by one or more of the attacks described.

5.4 *k*-Same Thwarts Face Recognition Software

Fig. 7a displays visual results of *k*-Same-Pixel (top row) and *k*-Same-Eigen (bottom row) for k= 1, 2, 3, 5, 10, 50, and 100 from left to right. The performance of naïve recognition, which matches original (gallery) to altered faces de-identified by *k*-Same-Pixel and *k*-Same-Eigen (probe), is provided in Fig. 7b. Both algorithms were tested for $k = 2$, $3, 5, 10, 50$, and 100, with person-specific face sets of 805 face images. Results are based on best match ("top rank"). The averaged results are plotted along with the expected value at each k. Correct recognition results from *k*-Same-Pixel and *k*-Same-Eigen experiments appear below the ideal value of $1/k$, which is also plotted as the top curve. The expected value, $1/k$, and the actual performance of *k*-Same-Pixel and *k*-Same-Eigen have an average difference of -1.6 percent (each) at $k = 2$ and -63 percent and -81 percent, respectively, at $k = 100$. The *k*-Same-Pixel and *k*-Same-Eigen results have an average absolute difference of 0.0017, compared on the same data set for 20 runs. The average absolute difference across the two different data sets is found to be 0.0050, which is larger than that between the two algorithms. Similar recognition results of less than $1/k$ were found with reverse recognition and parrot recognition.

These experimental results demonstrate *k*-Same-Pixel and *k*-Same-Eigen's ability to provide k-anonymity protection, and in so doing, thwart face recognition software from reliably recognizing their de-identified faces. Therefore

k-Same, BlackOut, and additive random noise (at approximately 13,000 pixels changed) are equally *effective* with respect to providing no better than 0.5 percent (random guessing) correct recognition. Of these, k-Same is *preservative* with respect to the Euclidean distance from the original image.

6 Related Work in Privacy Preserving Data Mining

This work can be viewed as privacy preserving data mining in high-dimensional data. Research in privacy preserving data mining techniques has been split into: secure multiparty computation, rule hiding, and perturbation techniques. Here is how face de-identification fits in. Face de-identification seeks to protect data that must be shared outside the original collecting institution, so secure multiparty computation has no bearing.

The goal in rule hiding [4], [18], [19] is to prevent sensitive rules from being revealed in shared data. One way to determine sensitive information for rule hiding in facial images might be to consider the principal components of the eigenface decomposition. If rules are considered as the eigenvectors of decomposition, then controlling for certain eigenvectors, through a technique such as suppressing them, may protect the identity of certain faces. However, different principal components, or a combination of such, can be sensitive for different faces and one cannot suppress all principal components. It remains a challenge to the research community to classify the features of an image into sensitive and nonsensitive rules for rule hiding techniques.

Perturbation approaches to privacy preserving data mining have been addressed in previous research, namely, [1], [2], [17]. In de-identifying faces, one could consider the set of original faces as X, and the set of perturbed faces as Y. The problem with such an approach in facial images is that modern face recognition software is highly insensitive to noise in the images. This was demonstrated in the experimental results reported in the previous section by additive random noise on individual pixels of the original face images. Additive random noise has little effect on the ability to dampen the recognition of de-identified faces by Eigenfaces, until roughly half of the pixel values have been flipped in black and white images or three-quarters of the pixels are changed to random values in gray-scale images. Facial details remaining in the de-identified images may appear horribly obscured due to the amount of perturbation necessary to sufficiently thwart recognition. There may exist different types of perturbation beyond additive random noise that would provide reasonable face de-identification. Using more complex perturbation models for face de-identification is an open research question.

7 Privacy Implications

Consider the real-world task of de-identifying a surveillance video clip. There is a preprocessing step to convert facial representations in the video clip to a person-specific face set. There is also a postprocessing step to place de-identified faces into a video clip so that the behaviors and actions of the people remain from the original video clip, but the de-identified faces replace the original facial representations. For example, a person coughing in the original video clip should be a de-identified person coughing in the final video clip. If needed, postprocessing can also apply encrypted identifiers to the de-identified faces, so that given a proper key, the original face for a person is revealed. Both the preprocessing and postprocessing steps contain active areas of research in computer vision, computer graphics and face recognition.

The basis of privacy protection of k-same algorithms is k-anonymity. There are numerous possible attacks on k-anonymity with known solutions [21]. One of the biggest problems is determining the proper privacy constraint k. The selection of k depends in part on whether protection must also prohibit the ability for all k individuals to be known. For example, for a k-samed face, all k subjects may be identified by name. Even though determining which named person appears in which frames in the de-identified video clip cannot be done exactly, the k people can all be identified and acted upon as a group. With other kinds of data, k is prescribed by policy, but even still, care must be taken.

De-identification, as described in this work, is not a guarantee of anonymity. This work seeks to thwart face recognition for the purpose of limiting automatic persistent recognition of populations whose images are captured on video but who have done nothing suspicious. While these techniques thwart face recognition software, for example, correct identification is still possible by recognizing clothes, behavior, or cooccurrences with other people.

Acknowledgements

The authors thank the anonymous viewers, who provided invaluable comments in improving the paper, and Ralph Gross, Granger Morgan, and Yiheng Li for insightful discussions. This research was sponsored in part by US Defense Advanced Research Projects Agency (DARPA) contract N00024-98-D-8124.

References

[1] D. Agrawal and C. Aggarwal, "On the Design and Quantification of Privacy Preserving Data Mining Algorithms," *Proc. ACM Symp. Principles of Database Systems,* pp. 247-255, 2001.

[2] R. Agrawal and S. Ramakrishnan, "Privacy-Preserving Data Mining," *Proc. ACM SIGMOD,* pp. 439-450, 2000.

[3] A.J. Allin, C.G. Atkeson, H. Wactlar, S. Stevens, M.J. Robertson, D. Wilson, J. Zimmerman, and A. Bharucha, "Toward the Automatic Assessment of Behavioral Disturbances of Dementia," *Proc. Fifth Int'l Conf. Ubiquitous Computing (UbiComp '03),* 2003.

[4] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. Verykios, "Disclosure Limitation of Sensitive Rules," *Proc IEEE Knowledge and Data Eng. Workshop (KDEX),* pp. 45-72, 1999.

[5] D. Blackburn, J. Bone, and P. Phillips, "Face Recognition Vendor Test (FRVT) 2000 Evaluation Report," NIST, Bethesda, Md., technical report, http://www.frvt.org/FRVT2000/documents.htm, 2001.

[6] Y. Cai, "Visual Privacy," presented at Topics in Privacy, Carnegie Mellon Univ., Pittsburgh, Pa, 2003.

[7] R. Gross, J. Cohn, and J. Shi, "Quo Vadis Face Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2001.

[8] Institute for Applied Autonomy, iSee Project: Survey and Web-Based Application Charting the Locations of Closed-Circuit Television (CCTV) Surveillance Cameras in Urban Environments, http://www.appliedautonomy.com/isee/info2.html, 2003.

[9] M. Jones, "All Eyes Are on Oceanfront's New Surveillance System," *The Virginian-Pilot,* 10 Sept. 2002.

[10] J. Lampinen and E. Oja, "Distortion Tolerant Pattern Recognition Based on Self-Organizing Feature Extraction," *IEEE Trans. Neural Networks,* vol. 6, no. 3, pp. 539-547, May 1995.
[11] H. Moon and P. Phillips, "Computational and Performance Aspects of PCA-Based Face Recognition Algorithms," *Perception,* vol. 30, no. 3, pp. 303-321, Mar. 2001.
[12] E. Newton, L. Sweeney, and B. Malin, "Preserving Privacy by De-Identifying Facial Images," Carnegie Mellon Univ., School of Computer Science, Pittsburgh, Pa, Technical Report CMU-CS-03-119, Mar. 2003.
[13] A. Oganian and J. Domingo-Ferrer, "On the Complexity of Microaggregation," *Proc. Joint Economic Commission for Europe, Work Session on Statistics Data Confidentiality,* 2001.
[14] P. Phillips, H. Moon, S. Rizvi, and P. Rauss, "The FERET Evaluation Methodology for Face-Recognition Algorithms," *IEEE Trans. Pattern Recognition and Machine Intelligence,* vol. 22, no. 10, pp. 1090-1104, Oct. 2000.
[15] P. Phillips, H. Wechsler, J. Huang, and P. Rauss, "The FERET Database and Evaluation Procedure for Face Recognition Algorithms," *Image and Vision Computing J.,* vol. 16, no. 5, pp. 295-306, Mar. 1998.
[16] J. Reeves, "Tampa Gets Ready for Its Closeup," *Time,* 16 July 2001.
[17] S. Rizvi and J. Haritsa, "Maintaining Data Privacy in Association Rule Mining," *Proc. 28th Conf. Very Large Data Base (VLDB '02),* 2002.
[18] Y. Saygin, V. Verykios, and C. Clifton, "Using Unknowns to Prevent Discovery of Association Rules," *SIGMOD Record,* vol. 30, no. 4, pp. 45-54, Dec. 2001.
[19] Y. Saygin, V. Verykios, and A. Elmagarmid, "Privacy Preserving Association Rule Mining," *Proc. 12th Int'l Workshop Research Issues in Data Eng. (RIDE),* 2002.
[20] L. Sirovich and M. Kirby, "Low-Dimensional Procedure for the Characterization of Human Faces," *J. Optical Soc. of Am.,* vol. 4, pp. 519-524, 1987.
[21] L. Sweeney, "*k*-Anonymity: A Model for Protecting Privacy," *Proc. Int'l J. Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 10, no. 5, pp. 557-570, 2002.
[22] L. Sweeney, "Report on the Potential of Early Detection Bioterroism Surveillance Using Non-Traditional Sources," presented at the Bio-ALIRT Program, Defense Advanced Research Projects Agency, Washington, DC, 2003.
[23] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neuroscience,* vol. 3, no. 1, pp. 71-86, 1991.
[24] B. Wall, "Digitizing Facial Features Fails to Prevent Identification of Plaintiff," *USA Today,* 10 Mar. 1996.
[25] E.W. Weisstein, *CRC Concise Encyclopedia of Mathematics,* second ed. Boca Raton, Fla.: CRC Press, 2002.

Elaine M. Newton is a fourth year doctoral student in the School of Engineering and Public Policy at Carnegie Mellon, as well as an adjunct researcher at RAND. Her research has focused on developing technology and policy that preserve reasonable anonymity in public spaces. At RAND, she coauthored *Army Biometric Applications: Identifying and Addressing Sociocultural Concerns* and worked with DARPA's Human ID at a Distance Program. Her Web page is http://www.andrew.cmu.edu/~enewton. She is a student member of the IEEE.

Latanya Sweeney received the PhD degree in computer science from MIT in 2001. She is a professor in the School of Computer Science at Carnegie Mellon University and director of the Data Privacy Lab. Her research interests are in privacy technology and in learning systems. She has received numerous awards, including several from AMIA, NLM, and APA, for her work on medical privacy technology. She is a member of the IEEE Computer Society and has been appointed to the IEEE USA Medical Technology Policy Committee. Her Web page is at http://privacy.cs.cmu.edu/people/sweeney/index.html.

Bradley Malin received the BS degree in biological sciences (2000) and the MS degree in knowledge discovery and data mining from Carnegie Mellon University (2002). He is currently working toward his doctorate in the Institute for Software Research International at Carnegie Mellon University. His research interests include data privacy, methods of reidentification, data mining, digital rights management, and bioinformatics. He has been a researcher in the Data Privacy Laboratory since its inception. He is a student member of the IEEE.