Face De-identification with Perfect Privacy Protection

Lily Meng*, Zongji Sun
School of Engineering and Technology
University of Hertfordshire
Hatfield AL10 9AB, UK
E-mail address:1.1.meng@herts.ac.uk

***Abstract*—The rising concern for privacy protection and the associated legal and social responsibilities have led to extensive research into the field of face de-identification over the last decade. To date, the most successful algorithms developed for face de-identification are those based on the *k*-Same de-identification, which guarantee a recognition rate lower than 1/*k*. However, the current *k*-Same solutions such as *k*-Same-Eigen and *k*-Same-M all rely on a decent value of *k* to deliver a good privacy protection. This paper proposes a departure from a fundamental aspect shared by the current *k*-Same solutions and thereby introduces a new member to the family which achieves perfect privacy protection for any original face regardless of the value of *k*.**

Keywords—privacy protection, face de-identification, active appearance model, k-Same, k-anonymity.

I. INTRODUCTION

Recent advances in both camera technology and computing hardware have highly facilitated the effectiveness and efficiency of image and video acquisition. This capability is now widely used in a variety of scenarios to capture images of people in target environments, either for immediate inspection or for storage and subsequent analysis/sharing [1]. These improved recording capabilities, however, have ignited concerns about the privacy of people identifiable in the scenes. The Council of Europe Convention of 1950 formally declared privacy protection as a human right. This was later embodied in the 1995 Data Protection Directive of the European Union (Directive 95/46/EC), which demands the deployment of appropriate technical and organizational measures to protect private information in the course of transferring or processing such data. This legal requirement along with ethical responsibilities has restricted data sharing and utilization while various organizations may require the use of such data for research, business, academic, security and many other purposes. To comply with the regulations, de-identification has become the focus of attention by many organizations with the ultimate goal of removing all personal identifying information while protecting the utility of the data.

Various methods have been proposed for the de-identification of faces in still and moving images. These methods can be put into two categories: the ad hoc methods (such as masking, pixelation and blurring [2-4] and the *k*-anonymity based methods (such as *k*-Same proposed by Newton et al. [5]). The ad hoc methods are usually simple to implement. However, they fail to serve their purpose as they are unable to thwart the existing face recognition software [5, 6]. To achieve privacy protection, the concept of *k*-anonymity was introduced by Sweeney in 2002 [7]. All *k*-anonymity based methods de-identify *k* original samples with an identical aggregate sample and hence achieve privacy protection with a recognition rate guaranteed to be lower than 1/*k*. In the field of face de-identification, the most widely used family of *k*-anonymity algorithms is *k*-Same [5]. However, all existing *k*-Same solutions share a drawback which makes them all produce a recognition rate just below the theoretical maximum of $1/k$ while the new *k*-Same solution proposed in this paper achieves a recognition rate of zero.

The remainder of the paper is structured as follows. Section 2 reviews the *k*-Same framework and points out the drawback shared by all existing *k*-Same solutions. Section 3 describes a new *k*-Same solution and proves that this new solution can always achieve zero recognition rate. Section 4 evaluates the proposed algorithm's ability to protect privacy and compares it to that of the existing *k*-Same solutions. Finally, the findings of this work are summarized and further discussed in Section 5.

II. *K*-SAME DE-IDENTIFICATION

A. Definition of k-Same De-identification

For the purpose of comparison with Newton et al.'s paper on *k*-Same [5], the same notations are used in this paper. The definition of *k*-Same de-identification from their paper is repeated here.

Definition 1 *k*-Same (Definition 2.10 in [5]). 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 $\Gamma_d = f(\Gamma)$ 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.

Person-specific face set in Definition 1 means that the face set contains only one face image of each person. A de-identification method $f: \mathbf{H} \rightarrow \mathbf{H}_d$ is said to be **preserved** if it minimizes the information loss measured by a precision metric $loss(\Gamma, f(\Gamma))$. The *k*-Same de-identification guarantees to be **effective** with respect to the claim in Definition 1 since it adds *k* copies of each Γ_d to $\mathbf{H}_d$.

B. The Common Drawback of Existing k-Same Solutions

To achieve *k*-anonymity, *k*-Same de-identification de-identify each cluster of at least *k* original faces with the same aggregate face and hence the name *k*-Same. The core problem of *k*-Same de-identification is to find the optimal selection of faces from the original face set to form the clusters of *k* faces. In [5] it is claimed that "basing each aggregate face on a cluster

of homogeneous original faces minimizes information loss". This claim is correct. However, all existing *k*-Same de-identification solutions share a common mistake when deciding which face cluster of homogeneous faces to be selected to de-identify a given face. They all select a cluster of faces that are closest to the original, implying that the loss to be minimized or the information to be preserved is the identities of the original faces. Obviously, this conflicts with the ultimate goal of *k*-Same which is privacy protection.

The information that should be preserved in the de-identified faces is the various data utilities such as gender, age and expressions. This means that the correct choice of $loss(\Gamma, f(\Gamma))$ should be a data utility function measuring the category and/or the quantity of the target data utilities to be preserved. To date, the most cited attempt on integrating utility preservation into face de-identification has been *k*-Same-Select [8]. Two data utility classifiers, a gender classifier and an expression classifier, are adopted in [8] for the purpose of preserving data utility. The utility classifiers are used to partition the original face set into mutually exclusive data utility subsets. Since the resultant utility subsets contain more than k faces each (half of the set has the same gender for example), the problem then becomes how to further divide each data utility subset into clusters of k. The *k*-Same-Select algorithm proposed in [8] has the same drawback as all the other existing *k*-Same solution, as it uses the closest faces to de-identify each original face.

For the benefit of discussions in this paper, the *k*-Same solutions that de-identify an original face based on the faces that are closest to it are referred to as "*k*-Same-closest algorithms". Examples of *k*-Same-closest algorithms include *k*-Same-Pixels [5], *k*-Same-Eigen [5], *k*-Same-M [6] and *k*-Same-Select [8]. In fact, *k*-Same-closest algorithms are the worst *k*-Same solutions in terms of privacy protection. When no overlapping exists between any two clusters, the original face that is closest to the average/center of a cluster must lie within the cluster (see proof of Theorem 2 in the next section). The *k*-Same-closest algorithms uses the center of a cluster to de-identify the k faces in that cluster, meaning the algorithms will always lead to a recognition rate equal to the theoretical maximum of $1/k$. When overlapping exists between two clusters the center of a cluster can be closest to an original face from the overlapping cluster (such as that demonstrated in Fig. 1 where the inner cluster of $\{3,4,5\}$ is formed earlier than the outer cluster $\{1,2,6\}$ and the outer cluster has an average equal to a member of the inner cluster), giving *k*-Same-closest algorithms lucky escapes. However, when de-identification is performed to an original face instead of a face set the *k*-Same-closest algorithms will never escape from the theoretical maximum and always generate the worst privacy protection within the *k*-anonymity framework. Even with a face set, experimental results (Fig. 6) show that the recognition rate of faces de-identified by the *k*-Same-closest algorithms always stays synchronized with the theoretical maximum of $1/k$ and force the *k*-Same-closest algorithms to use large k's to achieve decent privacy protection. Large values of k will not only lead to the requirement of large image gallery for the de-identification process but also to the discrimination issue with the de-identified faces. After all, there are k copies of each de-identified face.

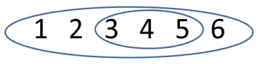

Fig. 1. Overlapping may occur between clusters of homogeneous faces when formed by a *k*-Same-closest algorithm.

III. The Proposed Face De-identification

A. A Better k-Same Solution

This section presents a new approach to face de-identification, which repeats each de-identified face at least k times in the de-identified face set $\mathbf{H}_d$ and therefore is a *k*-Same solution. However, this new *k*-Same solution is fundamentally different to the existing *k*-Same-closest algorithms. To best serve its goal of privacy protection, the proposed solution performs clustering with the aim to maximize the removal/loss of identity information in the original faces. In contrast to *k*-Same-closest which de-identifies an original face with the aggregate of the closest face cluster to it, it de-identifies each original face $\Gamma_i \in \mathbf{H}$ with an aggregate face of a cluster that, identity-wise, is furthest away from it and is hence named *k*-Same-furthest.

This work calculates the average as the aggregate of k faces, although other measures can be used to perform the aggregation. The identity distance can be the pixel-wise Euclidean distance or the distance measured in a projected feature space such as the Eigenface space [9] or the feature space constructed by an Active Appearance Model (AAM) [10, 11]. As pointed out in [6], *k*-Same is appearance based, operating entirely in the image space. As a result, 'ghosting' artefacts tend to appear due to the misalignments of the k images involved, even when images are aligned based on a small number of facial landmarks (e.g. the corners of the eyes and the tip of the nose). To prevent ghosting' artefacts in the de-identified faces, this work performs averaging of faces in the feature space constructed by an AAM. In addition, the identity distance is measured in the feature space constructed by the same AAM such that construction/training of additional feature space(s) is avoided. Fig. 2 outlines the process flow of the proposed *k*-Same-furthest algorithm.

Algorithm: *k-Same-furthest*($\mathbf{H}$, k)

Inputs: Face set $\mathbf{H}$ and privacy constraint k, with $|\mathbf{H}| \geq 2k$
An Active Appearance Model AAM

Output: De-identified face set $\mathbf{H}_d$ and its AAM projection $\mathbf{M}_d$

Uses: a face cluster $\mathbf{C}_c = \{\Lambda_{ci}\}$ with a center at $\overline{\Lambda_c}$ and a radius of r_c, a face cluster $\mathbf{C}_f = \{\Lambda_{fi}\}$ with a center at $\overline{\Lambda_f}$ and a radius of r_f, and $\text{dist}(\overline{\Lambda_c}, \overline{\Lambda_f})$ which is the distance between $\overline{\Lambda_c}$ and $\overline{\Lambda_f}$.

Steps:

1 $\mathbf{H}_d = \boldsymbol{\phi}$
2 $\mathbf{M} = \text{AAM}(\mathbf{H})$
3 `For` each $\Lambda_i \in \mathbf{M}$ `do`:
4 `If` $|\mathbf{M}| \geq 2k$ `then`:
5 Add Λ_i to $\mathbf{C}_c$ and remove it from $\mathbf{M}$

6 $\overline{\Lambda_c} = \Lambda_i, r_c = 0$
7 Select from $\mathbf{M}$ the face Λ_{f1} that is furthest away from Λ_i
8 Add Λ_{f1} to $\mathbf{C}_f$ and remove it from $\mathbf{M}$
9 $\overline{\Lambda_f} = \Lambda_{f1}, r_f = 0$
10 `While` $|\mathbf{C}_c| < k$ `and` $|\mathbf{C}_f| < k$ `do`
11 Select from $\mathbf{M}$ the face Λ_f that is closest to $\overline{\Lambda_f}$
12 Add Λ_f to $\mathbf{C}_f$
13 Update $\overline{\Lambda_f}$, r_f and dist$(\overline{\Lambda_c}, \overline{\Lambda_f})$
14 `If` dist$(\overline{\Lambda_c}, \overline{\Lambda_f}) \le r_c + r_f$ `then`
15 Remove Λ_f from $\mathbf{C}_f$
16 Update $\overline{\Lambda_f}$
17 `Break` from while loop
18 `Endif`
19 Remove Λ_f from $\mathbf{M}$
20 Select from $\mathbf{M}$ the face Λ_c that is closest to $\overline{\Lambda_c}$
21 Add Λ_c to $\mathbf{C}_c$
22 Update $\overline{\Lambda_c}$, r_c and dist$(\overline{\Lambda_c}, \overline{\Lambda_f})$
23 `If` dist$(\overline{\Lambda_c}, \overline{\Lambda_f}) \le r_c + r_f$ `then`
24 Remove Λ_c from $\mathbf{C}_c$
25 Update $\overline{\Lambda_c}$
26 `Break` from while loop
27 `Endif`
28 Remove Λ_c from $\mathbf{M}$
29 `Loop`
30 `If` $|\mathbf{C}_f| < k$ `then`
31 Select from $\mathbf{M}$ the closest $k - |\mathbf{C}_f|$ faces $\{\Lambda_{fi}\}$ to $\overline{\Lambda_f}$
32 Add $\{\Lambda_{fi}\}$ to $\mathbf{C}_f$, remove $\{\Lambda_{fi}\}$ from $\mathbf{M}$
33 `Endif`
34 `If` $|\mathbf{C}_c| < k$ `then`
35 Select from $\mathbf{M}$ the closest $k - |\mathbf{C}_c|$ faces $\{\Lambda_{ci}\}$ to $\overline{\Lambda_c}$
36 Add $\{\Lambda_{ci}\}$ to $\mathbf{C}_c$, remove $\{\Lambda_{ci}\}$ from $\mathbf{M}$
37 `Endif`
38 Add $\overline{\Lambda_f}$ to $\mathbf{M}_d$ to de-identify the faces in $\mathbf{C}_c$
39 Add $\overline{\Lambda_c}$ to $\mathbf{M}_d$ to de-identify the faces in $\mathbf{C}_f$
40 `else` // remaining original faces
41 `If` Λ_i is further to $\overline{\Lambda_f}$ than to $\overline{\Lambda_c}$ `then`
42 Add $\overline{\Lambda_f}$ to $\mathbf{M}_d$ to de-identify Λ_i
43 `Else`
44 Add $\overline{\Lambda_c}$ to $\mathbf{M}_d$ to de-identify Λ_i
45 `Endif`
46 `Endif`
47 `Next`
48 $\mathbf{H}_d = \text{AAM}^{-1}(\mathbf{M}_d)$

Fig. 2. The process flow of the proposed k-Same-furthest algorithm.

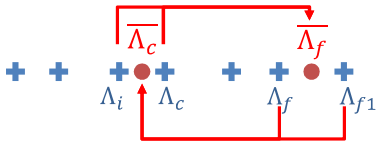

Fig. 3. Results of an iteration of the proposed k-Same-furthest de-identificaiton process on an example data set, where original samples $\Lambda_{f1}, \Lambda_{f2}$ are de-identified as Ψ_c (the average of Λ_{c1} and Λ_{c2}) and $\Lambda_{c1}, \Lambda_{c2}$ are de-identified as Ψ_f (the average of Λ_{f1} and Λ_{f2}). Here $k = 2$.

For a given face set $\mathbf{H}$, the proposed k-Same-furthest projects $\mathbf{H}$ to the feature space constructed by a trained AAM (line 2) and generates $\mathbf{M}$ which is the AAM representation of the original face set. All the subsequent operations of de-identification are performed in the AAM space, i.e. on $\mathbf{M}$. Lines 5 through 39 in Fig. 2 define an iteration of the proposed de-identification process when there are more than $2k$ faces remaining in $\mathbf{M}$. In each iteration the proposed k-Same-furthest algorithm de-identifies $2k$ original faces. Fig. 3 depicts the results of a de-identification iteration for an example data set when $k = 2$. To simplify illustration and ease understanding, a data set of scalars is used in Fig. 3. In Figs. 2 and 3, Λ_i is the original face that triggers the de-identification process (line 3 in Fig. 2) and Λ_{f1} is the furthest face to Λ_i (line 7). The cluster $\mathbf{C}_c$ consists of Λ_i and another face Λ_c. The cluster $\mathbf{C}_f$ consists of Λ_{f1} and Λ_f. Cluster $\mathbf{C}_c$ is formed by selecting from $\mathbf{M}$ (the remaining original faces) the closest faces to Λ_i (lines 20 and 21) and hence the closest cluster of faces to Λ_i that is available from $\mathbf{M}$. Cluster $\mathbf{C}_f$ is formed by selecting from $\mathbf{M}$ the closest faces to Λ_{f1} (lines 11 and 12) and hence the furthest cluster of faces to Λ_i that is available from $\mathbf{M}$. Cluster $\mathbf{C}_c$ may not be the closest cluster of faces to Λ_i in the full face set but it is the closest among the remaining original faces that have not been de-identified or added to a cluster. The same holds for cluster $\mathbf{C}_f$. To cause identity loss, the proposed k-Same-furthest de-identifies the members in a cluster with the center of the other cluster (Fig. 3 and lines 38 and 39 in Fig. 2). Since cluster $\mathbf{C}_c$ is identify-wise the closest cluster to Λ_i and $\mathbf{C}_f$ the furthest, they are identity-wise the furthest away from each other, meaning the proposed algorithm achieves maximum identity loss possible.

To avoid members of a cluster become the closest original to the center of the other (i.e. the situation illustrated in Fig. 1), overlapping must be avoided between $\mathbf{C}_c$ and $\mathbf{C}_f$. Whenever a new member is added to a cluster, k-Same-furthest checks to see whether overlapping is caused by this new member (lines 14 and 23). If so, this new member is removed from the cluster and the clustering loop for both $\mathbf{C}_c$ and $\mathbf{C}_f$ is stopped, as this new member is the closest to the cluster and therefore adding any other remaining face to $\mathbf{C}_c$ or $\mathbf{C}_f$ would cause even more overlapping between the two clusters. If clustering stopped before $\mathbf{C}_c$ and $\mathbf{C}_f$ has been assigned k faces each, faces closest to the center of each cluster are selected and added to the cluster to fill up the gaps. However, the centers of $\mathbf{C}_c$ and $\mathbf{C}_f$ are calculated using only those members that are added to them by the clustering process (i.e. before overlapping appears).

Each iteration defined by lines 5 to 39 de-identifies $2k$ faces. When $|\mathbf{H}|$ is not a multiple of $2k$, there will be fewer than $2k$ faces remaining in $\mathbf{M}$ (i.e. line 40 is true) after $|\mathbf{H}|/(2k)$ iterations. When this is the case, the operations defined by lines 41 to 45 are performed on each of the remaining faces. For a given remaining face from $\mathbf{M}$, line 41 identifies which of the last formed two face clusters is further to it. Then it is de-identified as the center of the further cluster (line 42 or 44), generating the maximize identity loss.

It is assumed in this work that there is no specific data utility to be preserved by the de-identification process. If there is a simultaneous requirement on the preservation of data utility, extra steps must be taken and there are two options. Option one is the approach adopted in [10], which involves the use of a data utility classifier. The original face set is first partitioned into subsets using the data utility classifier. The k-Same-furthest process is then performed on each utility subset. Option two performs k-Same-furthest face de-identification without consideration of data utility preservation and then restores the lost utility on the de-identified face generated by k-Same-furthest. The second option has been tested by the work presented in [12].

B. Correctness of the k-Same-furthest Algorithm

Theorem 1. If $\mathbf{H}$ is a person-specific face set, k is a privacy constraint, $\mathbf{H}_d = k\text{-Same-furthest}(\mathbf{H}, k)$, $k > 1$, and $|\mathbf{H}| \geq 2k$, then $\mathbf{H}_d$ satisfies k-anonymity.

Proof. Fig. 2 contains pseudo code for k-Same-furthest(). The proposed algorithm de-identifies original faces as the center of various face clusters. For each cluster center calculated, k (when $|\mathbf{H}|$ is a multiple of $2k$) or more (when $|\mathbf{H}|$ is not a multiple of $2k$) copies of the same center are added to $\mathbf{H}_d$, making the k or more copies in $\mathbf{H}_d$ indistinguishable. Or in other words, $\mathbf{H}_d$ satisfies k-anonymity and always guarantees a recognition rate less than $1/k$. **Theorem 1 is proved.**

Furthermore, lines 38, 39, 42 and 44 ensure that each copy of a cluster center in $\mathbf{H}_d$ has a one-to-one correspondence to an original face in $\mathbf{H}$. Therefore, 1) $|\mathbf{H}_d| = |\mathbf{H}|$; and 2) for each original face in $\mathbf{H}$ there exists a de-identified face in $\mathbf{H}_d$.

Theorem 2. Let $\overrightarrow{C_n}$ be the average of a cluster $\mathbf{C}^n$ of n faces and $\overrightarrow{C_{n+1}}$ the average of $\mathbf{C}^{n+1}$ which consists of $\mathbf{C}^n$ and a face $\vec{x} \in \mathbf{H}$. $\mathbf{C}^n \cap \mathbf{H} = \emptyset$. If $\vec{x}$ is the closest face in set $\mathbf{H}$ to $\overrightarrow{C_n}$, there cannot exit any other face in $\mathbf{H}$ that is closer to $\overrightarrow{C_{n+1}}$ than $\vec{x}$.

Proof. Because $\overrightarrow{C_{n+1}} = \frac{1}{n+1}\left(\vec{x} + n \cdot \overrightarrow{C_n}\right)$, $\overrightarrow{C_n}$, $\overrightarrow{C_{n+1}}$ and $\vec{x}$ are points on the same line in the feature space where $\mathbf{H}$ is defined, and $\overrightarrow{C_{n+1}}$ lies between $\overrightarrow{C_n}$ and $\vec{x}$. Fig. 4 illustrates the geometric relationships among $\overrightarrow{C_n}$, $\overrightarrow{C_{n+1}}$ and $\vec{x}$, assuming $\mathbf{H}$ is defined in a 2D space.

Let $\mathbf{S}^n$ be a sphere with a center at $\overrightarrow{C_n}$ and a radius of $\left|\overrightarrow{C_n} - \vec{x}\right|$ and $\mathbf{S}^{n+1}$ be a sphere with a center at $\overrightarrow{C_{n+1}}$ and a radius of $\left|\overrightarrow{C_{n+1}} - \vec{x}\right|$. Since $\overrightarrow{C_n}$, $\overrightarrow{C_{n+1}}$ and $\vec{x}$ are on the same line with $\overrightarrow{C_{n+1}}$ lying between $\overrightarrow{C_n}$ and $\vec{x}$, hence $\mathbf{S}^{n+1} \subset \mathbf{S}^n$. $\mathbf{S}^n$ and $\mathbf{S}^{n+1}$ are the solid and dotted circles in Fig. 4, respectively.

Let $\vec{y} \in \mathbf{H}$ be any other face than $\vec{x}$ (from set $\mathbf{H}$). Since $\vec{x}$ is the closest face in set $\mathbf{H}$ to $\overrightarrow{C_n}$, hence $\vec{y} \notin \mathbf{S}^n$. Because $\mathbf{S}^{n+1} \subset \mathbf{S}^n$, also hence $\vec{y} \notin \mathbf{S}^{n+1}$, meaning $\vec{y}$ is further away from $\overrightarrow{C_{n+1}}$ than $\vec{x}$. **Theorem 2 is proved.** Note that there can be faces within C^n that are closer to $\overrightarrow{C_{n+1}}$ than $\vec{x}$.

Theorem 3. If $\mathbf{H}$ is a person-specific face set, k is a privacy constraint, $k > 1$, $|\mathbf{H}| \geq 2k$, $\mathbf{H}_d = k\text{-Same-furthest}(\mathbf{H}, k)$, k-Same-furthest$(\mathbf{H}, k)$ uses $\mathrm{dist}(\Gamma_1, \Gamma_2)$ to measure the identity distance between any two faces Γ_1 and Γ_2, and $\Gamma_d \in \mathbf{H}_d$, there cannot exist any face recognition software that measures identity distance with $\mathrm{dist}(\Gamma_1, \Gamma_2)$ to correctly match Γ_d with the subject of its original face in $\mathbf{H}$.

Proof. Since $\mathbf{C}^{n+1}$ is composed of $\mathbf{C}^n$ and $\vec{x}$ and Theorem 2 states that no other face in $\mathbf{H}$ can be closer to $\overrightarrow{C_{n+1}}$ than $\vec{x}$, the face that is closest to $\overrightarrow{C_{n+1}}$ must be a member of $\mathbf{C}^{n+1}$. In other words, the face that is closest to the average of a cluster must be a member of that cluster when clusters are formed in the way described in Theorem 2, i.e. selecting the face that is to closest the current average and adding it to the cluster. This is exactly the way how k-Same-furthest() forms its clusters (lines 11-12 and 20-21). However, k-Same-furthest() forms two clusters simultaneously, meaning $\mathbf{C}^n$ in Theorem 2 is the union of $\mathbf{C}_c$ and $\mathbf{C}_f$ in k-Same-furthest(). A member in $\mathbf{C}_f$ can only be the closest face to the average of $\mathbf{C}_c$ when there is overlapping between $\mathbf{C}_f$ and $\mathbf{C}_c$, and vice versa. Since k-Same-furthest() (lines 14 and 23 in Fig. 2) ensures that no overlapping is allowed between $\mathbf{C}_c$ and $\mathbf{C}_f$, among the remaining original faces the face that is closest to the average of $\mathbf{C}_c$ must be a member of $\mathbf{C}_c$ according to Theorem 2, and vice versa. As k-Same-furthest() never de-identifies the members of a cluster as the average of that cluster, the de-identified faces (average of this cluster) can never be matched with their corresponding original faces (faces rather than members of this cluster), as long as the matching process uses the same distance measure as k-Same-furthest(). **Theorem 3 is proved.** Theorem 3 means that as long as the recognition software uses the same distance measure as itself, k-Same-furthest() guarantees to thwarts face recognition software for every face in its $\mathbf{H}_d$ and therefore the best k-Same solution in terms of privacy protection.

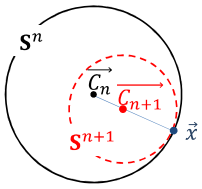

Fig. 4. Illustration of Theorem 2 in a 2D space.

Fig. 5. Example face images from the IMM dataset

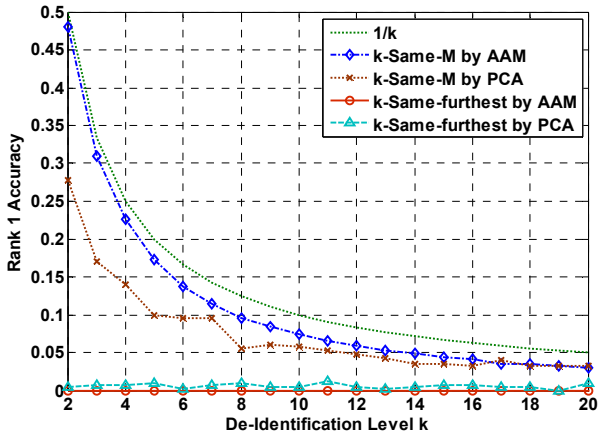

Fig. 6. Recognition rates for de-identified faces aginst original.

IV. EXPERIMENTS

A. Dataset

Experiments in this work were conducted with the IMM dataset [13], which contains images of 40 subjects. Only images with a near-frontal pose were used. These include a neutral, a happy and an arbitrary expression face images per subject. There is variation in head pose among the neutral as well as the happy faces. There is variation in both pose and lighting among the arbitrary expression faces. Fig. 5 shows some example face images from the IMM dataset.

B. Test Design

Privacy protection ability of the proposed *k*-Same-furthest algorithm is measured through recognition experiments using Eigenface technique and the Eigenface equivalent in the AAM space (i.e. that is used in *k*-Same-furthest). Cropped face images showing only the region inside the outline of the AAM-fitted shape are used in the experiments. 70% of the subjects are used for training the Eigenface or the AAM space with cropped original images. In testing, all cropped original images with various expressions are used as the gallery and the de-identified images as the probes. All results reported are based on randomly selecting ten different training and gallery/probe sets and computing the average recognition rate over all configurations.

C. *k*-Same-furthest Outperforms *k*-Same-closest

Fig. 6 shows the rank-1 recognition rates for the cropped original faces and the faces de-identified using either *k*-Same-M or the proposed *k*-Same-furthest. As shown in Fig. 6, when the same distance measure is used, *k*-Same-furthest always produces a recognition rate of zero. When PCA representation of face images is used the face recognition software, the recognition rates of the *k*-Same-furthest de-identified faces are slightly above zero whilst stay far lower than *k*-Same-M faces. When the face recognition software represents faces as AAM parameters, the recognition rates of the faces de-identified by *k*-Same-M remain around 2-3% below the theoretical maximum of $1/k$. When PCA representation is used by the face recognition software, the recognition rates of the *k*-Same-M faces is lower than those calculated by the AAM-based recognition but still stay much higher than the recognition rate achieved by *k*-Same-furthest. Regardless of which feature space the recognition software uses to represent face images, the recognition rate of the *k*-Same-M faces always stay synchronized with the theoretical maximum of $1/k$. This forces *k*-Same-M to use large k's in order to achieve decent privacy protection. With *k*-Same-furthest, the recognition rate stay at zero or slightly above zero, allowing decent privacy protection to be achievable with small *k*'s. This means that for a required level of privacy protection, *k*-Same-furthest requires a smaller *k* than *k*-Same-closest and in turn delivers a better discrimination among faces in the de-identified face set. After all, there are *k* copies of each face in the set.

Fig. 7 displays the visual results of the proposed *k*-Same-furthest algorithm with various values of *k* for three different expression faces of the same individual. Here expression of the original is preserved by forming clusters only with the original faces displaying the same expression. Since each original face in Fig. 7 is the first face to be de-identified, the same face will always be selected as Λ_{f1} (line 7 in Fig. 2) regardless of *k* and the selected Λ_{f1} is the furthest face to it over the entire face gallery. The cluster of faces that is used to de-identify it is the cluster of *k* faces that are closest to face Λ_{f1}. Since Λ_{f1} is the same regardless of *k*, nearly identical de-identified faces are generated by *k*-Same-furthest for each original face in Fig. 7 over various values of *k*. When the original face is not the first to de-identify the face that is furthest away from it over the entire face gallery might have already been taken by the previously formed clusters. As a result, a different Λ_{f1} and hence a different de-identified face will be generated for different *k* values.

Fig. 8 displays the visual results of the proposed *k*-Same-furthest algorithm where no preservation of data utility is implemented and hence the entire image gallery containing faces with various expressions and head poses is used to form the face clusters. The de-identified faces generated with various values of *k* display various expressions and head poses but a nearly identical identity. When *k* is small the de-identified faces tend to display an expression and a head pose of an extreme from the gallery. As *k* increases, the cluster on which the de-identified face is based becomes more and more diverse. As a result, the de-identified face converges to the average of the half gallery that is further to the original.

As displayed in Figs. 7 and 8, the de-identified faces for each original appear significantly different to their corresponding originals.

V. DISCUSSION AND CONCLUSION

This paper refers to *k*-Same-Pixel/-Eigen and all their current extensions as the *k*-Same-closest algorithms and has pointed out that they share the fundamental drawback of de-identifying faces based on faces closest to them. This means that the *k*-Same-closest algorithms are actually trying to minimize identity loss instead of maximising it and restricts the *k*-Same-closest algorithms to achieve decent privacy protection at the cost of large values of *k*, which in turn lead to the demand for a large image gallery or otherwise lack of discrimination among the de-identified faces (number of

distinctive faces in the de-identified face set is equal to or less than size of the gallery divided by k).

In contrast to k-Same-closest algorithms, the proposed k-Same-furthest algorithm de-identifies faces based on the faces that are furthest away from them and hence maximizes identity loss, achieving the perfect privacy protection regardless of the value of k.

REFERENCES

[1] L. Sweeney, Surveillance of Surveillances camera watch project, http://dataprivacylab.org/dataprivacy/projects/camwatch, 2005.

[2] J. Crowley, J. Coutaz, F. Berard, "Things that see," Communications of the ACM, vol. 43, pp. 54–64, 2000.

[3] C. Neustaedter, and S. Greenberg, "Balancing privacy and awareness in home media spaces," Workshop on Ubicomp Communities: Privacy as Boundary Negotiation, in conjunction with the 5th Int'l Conf. Ubiquitous Computing (UBICOMP), Seattle, WA , 2003.

[4] M. Boyle, C. Edwards, and S. Greenberg, "The effects of filtered video on awareness and privacy," Proc. ACM Conf. Computer Supported Cooperative Work, 2000.

[5] E. M. Newton, L. Sweeney, and B. Malin, "Preserving privacy by de-identifying face images," IEEE Trans. Knowledge and Data Eng., vol. 12, no. 2, pp. 232 – 243, February 2005.

[6] R. Gross, L. Sweeney, F. de la Torre, and S. Baker, "Model-based face de-identification," IEEE Workshop on Privacy Research in Vision, 2006.

[7] L. Sweeney, "k-Anonymity: a model for protecting privacy," Int'l J. Uncertainty, Fuzziness, and Knowledge-Based Systems, vol. 10, no. 5, pp. 557–570, 2002.

[8] R. Gross, E. Airoldi, B. Malin, and L. Sweeney, "Integrating utility into face de-identification," Workshop on Privacy-Enhanced Technologies, 2005.

[9] M. Turk, and A. P. Pentland, "Eigenfaces for recognition," J. Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86, 1991.

[10] G. Edwards, C. Taylor, and T. Cootes, "Interpreting Face Images Using Active Appearance Models," Proc. FG'98, pp. 300–305, Apr. 1998.

[11] I. Matthews, and S. Baker, "Active appearance models revisited," Int'l J. Computer Vision, vol. 60, no. 2, pp. 135-164, Nov. 2004.

[12] L. Meng, Z.J. Sun, K.L. Bennett, and A. Ariyaeeinia, "Retaining Expressions on De-identified Faces," submitted to the 37th Intl. Convention MIPRO, Special Session on BiForD, Opatija, Croatia, May 2014.

[13] M.M. Nordstrøm, M. Larsen, J. Sierakowski, and M.B. Stegmann, "The IMM face database - an annotated dataset of 240 face images," Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, May 2004.

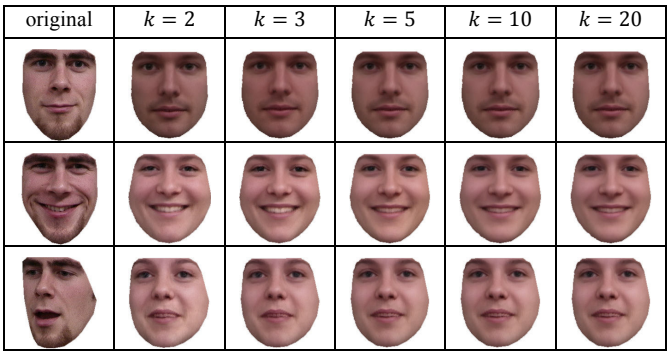

Fig. 7. Visual results of the proposed algorithm with expression of the original is preserved by forming clusters only with the original faces displaying the same expression.

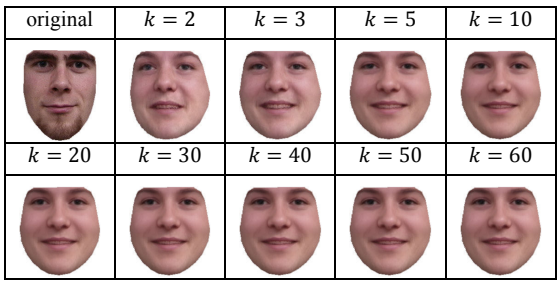

Fig. 8. Visual results of the proposed algorithm where the entire image gallery containing faces with various expressions and head poses is used to form the face clusters.