Face De-Identification

Ralph Gross, Latanya Sweeney, Jeffrey Cohn, Fernando de la Torre, and Simon Baker

Abstract With the emergence of new applications centered around the sharing of image data, questions concerning the protection of the privacy of people visible in the scene arise. In most of these applications knowledge of the identity of people in the image is not required. This makes the case for image de-identification, the removal of identifying information from images, prior to sharing of the data. Privacy protection methods are well established for field-structured data, however, work on images is still limited. In this chapter we review previously proposed naïve and formal face de-identification methods. We then describe a novel framework for the de-identification of face images using multi-factor models which unify linear, bilinear, and quadratic data models. We show in experiments on a large expression-variant face database that the new algorithm is able to protect privacy while preserving data utility. The new model extends directly to image sequences, which we demonstrate on examples from a medical face database.

1 Introduction

Recent advances in both camera technology as well as supporting computing hardware have made it significantly easier to deal with large amounts of visual data.

Ralph Gross · Latanya Sweeney
Data Privacy Lab, School of Computer Science, Carnegie Mellon University,
e-mail: {rgross,latanya}@cs.cmu.edu

Jeffrey Cohn
University of Pittsburgh e-mail: jeffcohn@pitt.edu

Fernando de la Torre
Robotics Institute, Carnegie Mellon University, e-mail: ftorre@cs.cmu.edu

Simon Baker
Microsoft Research, Microsoft Corporation, e-mail: sbaker@microsoft.com

This enables a wide range of new usage scenarios involving the acquisition, processing and sharing of images. However, many of these applications are plagued by privacy problems concerning the people visible in the scene. Examples include the Google Streetview service, surveillance systems to help monitor patients in nursing homes [3], and the collection and distribution of medical face databases (studying e.g. pain [1]).

In most of these applications knowledge of the identity of people in the image is not required. This makes the case for image de-identification, the removal of identifying information from images, prior to sharing of the data. Privacy protection methods are well established for field-structured data [30], however, work on images is still limited. The implicit goal of these methods is to protect privacy and preserve data utility, e.g. the ability to recognize gender or facial expressions from de-identified images. While initial methods discussed in the literature were limited to applying naïve image obfuscation methods such as blurring [23], more recent methods such as the k-Same algorithm provide provable privacy guarantees [14,25].

In this chapter we review the previously proposed naïve and formal face de-identification methods, highlighting their strengths and weaknesses (Section 2). The majority of previously introduced methods operate directly on image data which varies both with identity as well as non-identity related factors such as facial expressions. A natural extension of these methods would use a factorization approach to separate identity and non-identity related factors to improve preservation of data utility. However, existing multi-factor models such as the bilinear models introduced by Tenenbaum and Freeman [33] or tensor models [36] require complete data labels during training which are often not available in practice. To address this problem we describe a new multi-factor framework which combines linear, bilinear, and quadratic models. We show in experiments on a large expression-variant face database that the new algorithm is able to protect privacy while preserving data utility (Section 3). The new model extends directly to image sequences, which we demonstrate on examples from a medical face database (Section 4).

2 Related Work

The vast majority of previously proposed algorithms for face de-identification fall into one of two groups: ad-hoc distortion/surpression methods [7,10,18,23,24] and the k-Same [14,25] family of algorithms implementing the k-anonymity protection model [30]. We describe both categories of algorithms along with their shortcomings in this section.

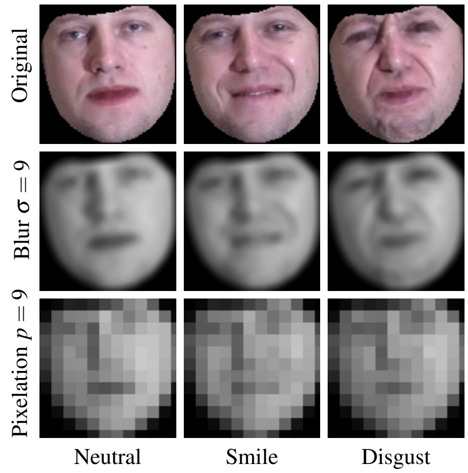


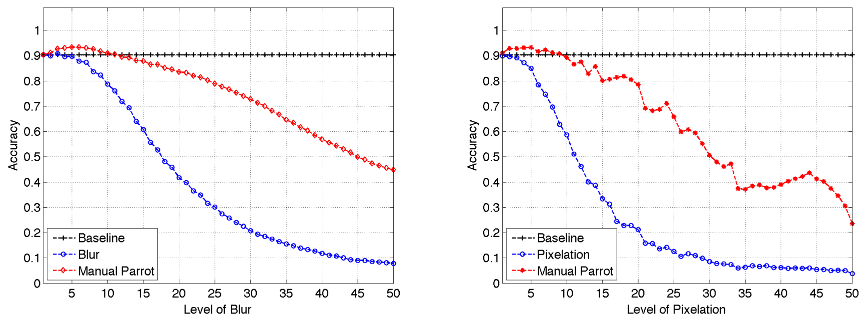
Fig. 1 Examples of applying the naïve de-identification methods blurring and pixelation to images from the CMU Multi-PIE database [15].

2.1 Naïve De-Identification Methods

Following similar practices in traditional print and broadcasting media, image distortion approaches to face de-identification alter the region of the image occupied by a person using data surpression or simple image filters. These ad-hoc methods have been discussed numerous times in the literature [7, 10, 18, 23, 24], often in the context of computer supported cooperative work (CSCW) and home media spaces where explicit user control is desired to balance between privacy and data utility.

Image filtering approaches use simple obfuscation methods such as blurring (smoothing the image with e.g. a Gaussian filter with large variance) or pixelation (image subsampling) [7, 24]. See Figure 1 for examples. While these algorithms are applicable to all images, they lack a formal privacy model. Therefore no guarantees can be made that the privacy of people visible in the images is actually protected. As a consequence naïve de-identification methods neither preserve privacy nor data utility as results presented in the next section show.

Work on privacy protection in video surveillance scenarios favors data suppression. Systems typically determine the region of interest in the image through varying combinations of standard computer vision techniques such as background subtraction [29, 35], object tracking [39], and in some cases face detection [27, 37]. Approaches proposed in the literature then differ in which area of the object to mask such as the face [8,21], the silhouette (thereby preserving the body shape) [20,31,38] or the entire bounding box covering the person [12, 38]. The amount of information transmitted can be further reduced by only indicating the position of the person in the image by e.g. a dot and replacing the image area occupied by the person in the original image with static background in the de-identified image [20]. An alterna-



(a) Blur (b) Pixelation

Fig. 2 Manual parrot recognition for both blurred and pixelated images. Instead of comparing de-identified images in the probe set with original images in the gallery we apply the same transformation to the gallery images prior to PCA recognition. This was termed *parrot* recognition by Newton et al. [25]. For both de-identification algorithms, recognition rates drastically improve, thereby demonstrating the vulnerability of both privacy protection algorithm to this attack.

tive approach in the video surveillance space which provides some user control on the amount of distortion applied in the image was proposed by Dufaux et al. [11]. Following background subtraction the regions of interest in the image are distorted by scrambling the coefficients used to encode the areas in a Motion JPEG 2000 [32] compressed video sequence. The magnitude of change is controlled by the number of altered coefficients.

2.2 Defeating Naïve De-Identification Methods

While naïve de-identification algorithms such as blurring and pixelation have been shown to successfully thwart human recognition [7,40] they lack an explicit privacy model and are therefore vulnerable to comparatively simple attacks. A very effective approach to defeat naïve de-identification algorithms was proposed by Newton et al. as (manual) *parrot recognition* [25]. Instead of comparing de-identified images to the original images (as humans implicitly or explicitly do), parrot recognition applies the same distortion to the gallery images as contained in the probe images prior to performing recognition. As a result, recognition rates drastically improve, in effect reducing the privacy protection offered by the naïve de-identification algorithms.

We demonstrate this empirically using frontal images from 228 subjects from the CMU Multi-PIE database [15] displaying *neutral*, *smile*, and *disgust* expressions. Images are shape normalized using manually established Active Appearance Model labels [9,22] (see Figure 1). We then build Principle Component Analysis [19] bases on a small subset of the data (68 subjects representing 30% of the data) and encode

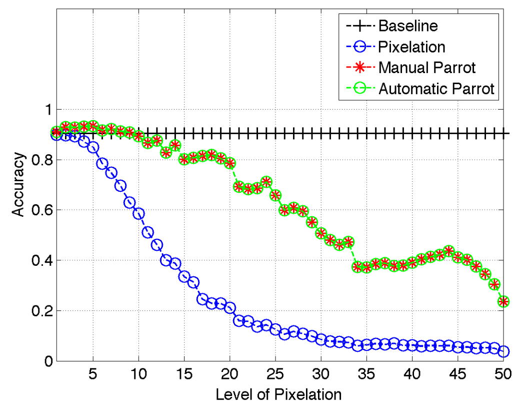


Fig. 3 Automatic parrot recognition for pixelation. We automatically determine the degree of pixelation applied to probe images by determining the size of blocks of equal (or approximately equal) pixel intensities in the image. The same pixelation is then applied to gallery images prior to PCA recognition. The resulting recognition accuracy is identical to the accuracy achieved if ground-truth knowledge of the correct pixelation degree is assumed.

the remainder of the data using these basis vectors. With the *neutral* face images as gallery and *smile* and *disgust* images of varying blur and pixelation levels as probes[1] we compute recognition rates using a whitened cosine distance, which has been shown to perform well in face PCA spaces [4]. In Figure 2 we compare accuracies for relating de-identified to original images with parrot recognition rates. For both blurring and pixelation, parrot recognition rates are significantly higher than the original de-identification rates. For low parameter settings of either algorithm, parrot recognition performs even better than using original, unaltered images in both gallery and probe. This is likely due to a reduction in image noise in de-identified images.

In the experiments for Figure 2 knowledge of the amount of blurring or pixelation present in the probe images was used to de-identify the gallery images with the same amount. This information, however, can be extracted directly from the de-identified probe images for an automatic parrot attack. In the case of pixelation we simply determine the size of blocks of equal (or approximately equal) pixel intensities in the image. As shown in Figure 3 the resulting recognition rates are identical to the manual procedure. A similar procedure can be applied in the case of blurring by analyzing the frequency spectrum of the de-identified images.

[1] The *gallery* contains images of known subjects. The probe images are compared against the gallery to determine the most likely match [26].

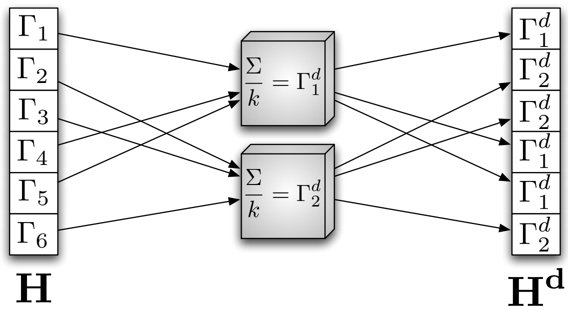


Fig. 4 Overview of the k-Same algorithm. Images are de-identified by computing averages over the closest neighbors of a given face in H and adding k copies of the resulting average to H^d.

2.3 *k*-Same

The k-Same family of algorithms [14,16,25] implement the k-anonymity protection model [30] for face images. Given a *person-specific*[2] set of images $H = \{\mathbf{I}_1, \ldots, \mathbf{I}_M\}$, k-Same computes a de-identified set of images $H^d = \{\mathbf{I}_1^d, \ldots, \mathbf{I}_M^d\}$ in which each $\mathbf{I}_i^d$ indiscriminately relates to at least k elements of H. It can then be shown that the best possible success rate for a face recognition algorithm linking an element of H^d to the correct face in H is $\frac{1}{k}$. See [25] for details. k-Same achieves this k-anonymity protection by averaging the k closest faces for each element of H and adding k copies of the resulting average to H^d. See Figure 4 for an illustration of the algorithm.

While k-Same provides provable privacy guarantees, the resulting de-identified images often contain undesireable artifacts. Since the algorithm directly averages pixel intensity values, even small alignment errors of the underlying faces lead to "ghosting" artifacts. See Figure 5(a) for examples. To overcome this problem we introduced a model-based extension to k-Same, referred to as k-Same-M in [16]. The algorithm fits an Active Appearance Model (AAM) [9,22] to input images and then applies k-Same on the AAM model parameters. The resulting de-identified images are of much higher quality than images produced by k-Same while the same privacy guarantees can still be made. See Figure 5(b) for examples.

k-Same selects images for averaging based on raw Euclidean distances in image space or Principal Component Analysis coefficient space [25]. In order to use additional information during image selection such as gender or facial expression labels we introduced k-Same-Select in [14]. The resulting algorithm provides k-anonymity protection while better preserving data utility. See Figure 6 for examples images from the k-Same and k-Same-Select algorithms.

The k-Same family of algorithms [14, 16, 25] implement the k-anonymity protection model [30] for face images. Given a *person-specific*[3] set of images $H = \{\mathbf{I}_1, \ldots, \mathbf{I}_M\}$, k-Same computes a de-identified set of images $H^d = \{\mathbf{I}_1^d, \ldots, \mathbf{I}_M^d\}$ in

[2] In a person-specific set of faces each subject is represented by no more than one image.

[3] In a person-specific set of faces each subject is represented by no more than one image.

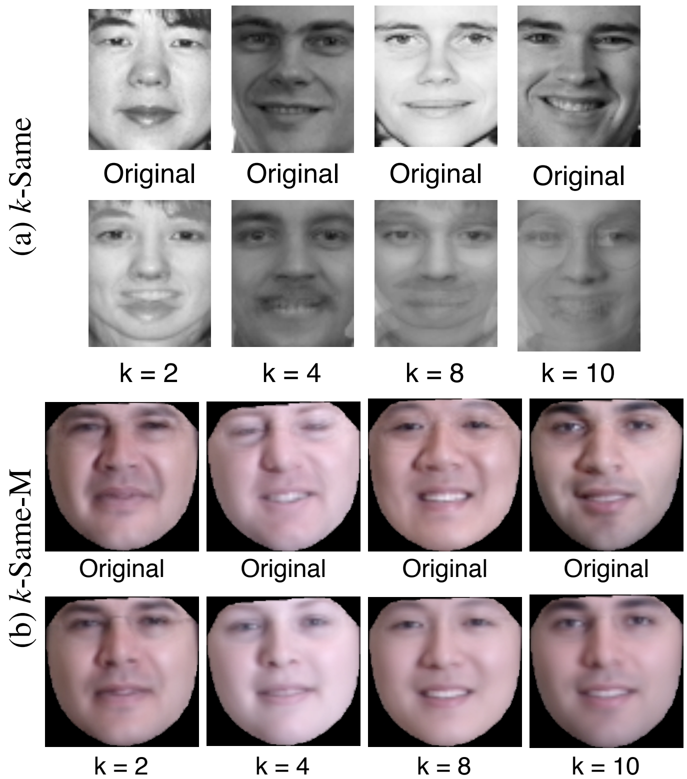


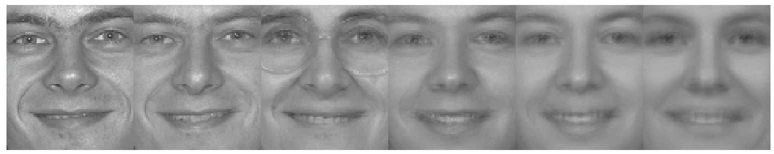
Fig. 5 Examples of de-identified face images. Faces shown in (a) were de-identified using the appearance-based version of *k*-Same. Due to misalignments in the face set, ghosting artifacts appear. Faces in (b) were de-identified using *k*-Same-M, the model-based extension of *k*-Same. In comparison, the images produced by *k*-Same-M are of much higher quality.

which each $\mathbf{I}_i^d$ indiscriminately relates to at least k elements of H. It can be shown that the best possible success rate for a face recognition algorithm linking an element of H^d to the correct face in H (independent of the algorithm used) is $\frac{1}{k}$. See [25] for details. k-Same achieves k-anonymity protection by averaging the k closest faces for each element of H and adding k copies of the resulting average to H^d. In order to use additional information during image selection such as gender or facial expression labels, k-Same-Select was introduced in [14]. The resulting algorithm provides k-anonymity privacy protection while preserving data utility as evidenced by both gender and facial expression recognition experiments.

While k-Same provides adequate privacy protection, it places strong restrictions on the input data. The algorithm assumes that each subject is only represented once in the dataset, a condition that is often not met in practice, especially in video sequences.

Original $k = 2$ $k = 3$ $k = 5$ $k = 10$ $k = 20$

(a) k-Same



Original $k = 2$ $k = 3$ $k = 5$ $k = 10$ $k = 20$

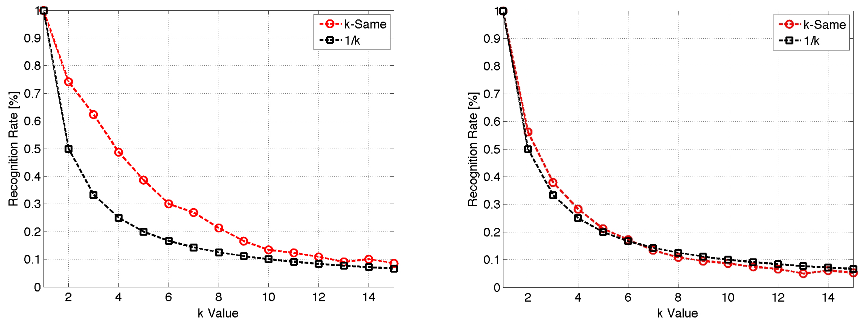
(b) k-Same-Select

Fig. 6 Examples of applying k-Same and k-Same-Select to expression variant faces. Since k-Same-Select factors facial expression labels into the image selection process, facial expressions are preserved better (notice the changing expression in the first row). Both algorithm provide k-anonymity privacy protection.

2.4 Shortcomings of the k-Same Framework

k-Same assumes that each subject is only represented once in the dataset H, a condition which is often not met in practice. Since k-Same uses the nearest neighbors of a given image during de-identification, the presence of multiple images of the same subject in the input set can lead to lower levels of privacy protection. To demonstrate this we report results of experiments on the Multi-PIE database [15]. Each face in the dataset is represented using the appearance coefficients of an Active Appearance Model [9, 22]. Recognition is performed by computing the nearest neighbors in the appearance coefficient space. In the first experiment we employ images of 203 subjects in frontal pose and frontal illumination, displaying neutral, surprise, and squint expressions. In the second experiment we use images of 249 subjects recorded in frontal pose, displaying neutral expressions. Images of five illumination conditions per subject are included in the dataset. In either case, k-Same fails to provide adequate privacy protection. Figure 7 shows face recognition accuracies for varying levels of k. For the expression-variant dataset, accuracies stay well above the $\frac{1}{k}$ rate guaranteed by k-Same for datasets with single examples per class (see Figure 7(a)). The same observation holds for the illumination-variant dataset for low k values (see Figure 7(b)). We obtain similar results even when class information is factored into the k-Same de-identification process. We can conclude that k-Same does not provide sufficient privacy protection if multiple images per subject are included in the dataset.

k-Same operates on a *closed* face set H and produces a corresponding de-identified set of faces H^d. Many potential application scenarios for de-identification techniques involve processing individual images or sequences of images. k-Same is



(a) k-Same on expression-variant face data (b) k-Same on illumination-variant face data

Fig. 7 Recognition performance of k-Same on image sets containing multiple faces per subject. (a) shows recognition accuracies after applying k-Same to a subset of the CMU Multi-PIE database containing multiple expressions (neutral, surprise, and squint) of each subject. (b) shows recognition accuracies after applying k-Same on an illumination-variant subset of Multi-PIE. For both datasets recognition accuracies exceed $\frac{1}{k}$, indicating lower levels of privacy protection.

not directly applicable in these situations. Due to the definition of k-Same, extensions for open-set de-identification are not obvious.

3 Multi-Factor Face De-Identification

To address the shortcomings of the k-Same framework described in Section 2.4 we proposed a multi-factor framework for face de-identification [13, 17], which unifies linear, bilinear, and quadratic models. In our approach we factorize input images into identity and non-identity factors using a generative multi-factor model. We then apply a de-identification algorithm on the combined factorized data before using the bases of the multi-factor model to reconstruct de-identified images. See Figure 8 for on overview. In the following we first define our unified model (Section 3.1). We then describe two fitting algorithms, the alternating and joint fitting algorithm and compare their performance on synthetic data (Section 3.2). In Section 3.3 we describe how to extend the model to include additional constraints on basis and coefficient vectors. We present results from an evaluation of the algorithm on a face de-identification task in Section 3.4.

3.1 Reconstructive Model

We define the general model M for data dimension k as

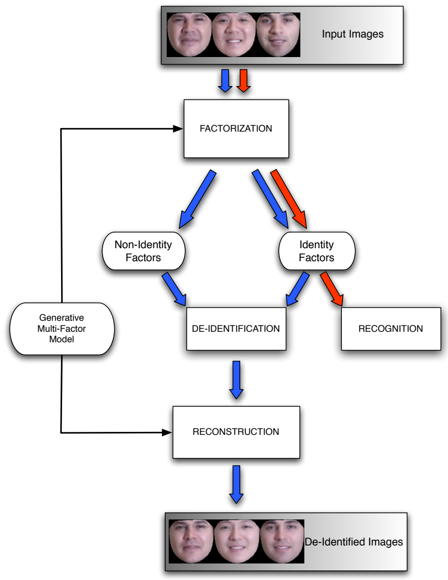


Fig. 8 Overview of the combined framework for face de-identification introduced in this thesis. Input images are factorized into identity and non-identity components using a generative multi-factor model. The resulting identity parameters could be used for face recognition (red arrows) or, together with the non-identity parameters for face de-identification (blue arrows). After de-identification, the bases of the multi-factor model are used to produce de-identified images.

$$M_k(\mu,\mathbf{B}^1,\mathbf{B}^2,\mathbf{W},\mathbf{Q}^1,\mathbf{Q}^2;\mathbf{c}_1,\mathbf{c}_2) = \\ \begin{pmatrix} 1 & \mathbf{c}_1^T & \mathbf{c}_2^T \end{pmatrix} \underbrace{\begin{pmatrix} \mu_k & \mathbf{B}_k^2 & 0 \\ \mathbf{B}_k^1 & \mathbf{W}_k & \mathbf{Q}_k^1 \\ 0 & \mathbf{Q}_k^2 & 0 \end{pmatrix}}_{\Omega_k} \begin{pmatrix} 1 \\ \mathbf{c}_2 \\ \mathbf{c}_1 \end{pmatrix} \tag{1}$$

with mean μ, linear bases $\mathbf{B}^1,\mathbf{B}^2$, bilinear basis $\mathbf{W}$, quadradic bases $\mathbf{Q}^1,\mathbf{Q}^2$, and coefficients $\mathbf{c}_1$ and $\mathbf{c}_2$. $\mathbf{c}^1 \in \mathbb{R}^{r_1}, \mathbf{c}^2 \in \mathbb{R}^{r_2}, \mu \in \mathbb{R}^d, \mathbf{B}^1 \in \mathbb{R}^{d\times r_1}$ with $\mathbf{B}_k^1 \in \mathbb{R}^{1\times r_1}$, $\mathbf{B}^2 \in \mathbb{R}^{d\times r_2}, \mathbf{W}_k \in \mathbb{R}^{r_1\times r_2}, \mathbf{Q}_k^1 \in \mathbb{R}^{r_1\times r_1}, \mathbf{Q}_k^2 \in \mathbb{R}^{r_2\times r_2}$. To avoid redundancy, $\mathbf{Q}^1,\mathbf{Q}^2$ could be either symmetric or upper triangular. Here we choose upper triangular.

While Eqn. (1) defines a quadratic model, it in fact contains lower-dimensional linear, bilinear and quadratic models as special cases. To illustrate this we set $\mathbf{W} = \mathbf{Q}^1 = \mathbf{Q}^2 = 0$ and obtain

$$
\begin{aligned}
M_k^{Lin}(\mu,\mathbf{B}^1,\mathbf{B}^2,0,0,0;\mathbf{c}_1,\mathbf{c}_2) &= \\
&= \begin{pmatrix}1 & \mathbf{c}_1^T & \mathbf{c}_2^T\end{pmatrix} \begin{pmatrix}\mu_k & \mathbf{B}_k^2 & 0 \\ \mathbf{B}_k^1 & 0 & 0 \\ 0 & 0 & 0\end{pmatrix} \begin{pmatrix}1 \\ \mathbf{c}_2 \\ \mathbf{c}_1\end{pmatrix} \\
&= \mu_k + \mathbf{c}_1^T\mathbf{B}_k^1 + \mathbf{B}_k^2\mathbf{c}_2
\end{aligned}
$$

the linear model in $\mathbf{c}^1$ and $\mathbf{c}^2$. Similarly, for $\mathbf{Q}^1 = \mathbf{Q}^2 = 0$ we obtain the bilinear model

$$
\begin{aligned}
M_k^{Bilin}(\mu,\mathbf{B}^1,\mathbf{B}^2,\mathbf{W},0,0;\mathbf{c}_1,\mathbf{c}_2) &= \\
&= \begin{pmatrix}1 & \mathbf{c}_1^T & \mathbf{c}_2^T\end{pmatrix} \begin{pmatrix}\mu_k & \mathbf{B}_k^2 & 0 \\ \mathbf{B}_k^1 & \mathbf{W}_k & 0 \\ 0 & 0 & 0\end{pmatrix} \begin{pmatrix}1 \\ \mathbf{c}_2 \\ \mathbf{c}_1\end{pmatrix} \\
&= \mu_k + \mathbf{c}_1^T\mathbf{B}_k^1 + \mathbf{B}_k^2\mathbf{c}_2 + \mathbf{c}_1^T\mathbf{W}_k\mathbf{c}_2
\end{aligned}
$$

Mixtures of the components yield model combinations, i.e. mixed linear and bilinear, mixed bilinear and quadratic, etc. The model as defined in Eqn. (1) is ambiguous, i.e. there exists transformations of $\mathbf{c}^1,\mathbf{c}^2$, and Ω_k that produce identical data vectors. See [17] for more details.

3.2 Model Fitting

The goal of fitting is to compute the parameters that minimize the model reconstruction error for a given training data set d $= [\mathbf{d}_1 \dots \mathbf{d}_n]$:

$$\arg\min_{\Gamma,\mathbf{C}_1,\mathbf{C}_2} \sum_{l=1}^{n} \|M(\Gamma;\mathbf{c}_1(l),\mathbf{c}_2(l)) - \mathbf{d}_l\|_2^2 \tag{2}$$

with the bases $\Gamma = (\mu,\mathbf{B}^1,\mathbf{B}^2,\mathbf{W},\mathbf{Q}^1,\mathbf{Q}^2)$ and coefficients $\mathbf{C}_1 = (\mathbf{c}_1(1),\dots,\mathbf{c}_1(n)), \mathbf{C}_2 = (\mathbf{c}_2(1),\dots,\mathbf{c}_2(n))$.

For the linear model M^{Lin} the corresponding minimization problem is

$$\arg\min_{\mathbf{B},\mathbf{C}} \sum_{l=1}^{n} \|M^{Lin}(\mathbf{B};\mathbf{c}(l)) - \mathbf{d}_l\|_2^2 \tag{3}$$

where we combined the separate bases $\mathbf{B}^1,\mathbf{B}^2$ into $\mathbf{B}$ and the coefficients $\mathbf{C}_1,\mathbf{C}_2$ into $\mathbf{C} = (\mathbf{c}(1),\dots,\mathbf{c}(n))$. Eqn. (3) can be minimized efficiently by using PCA (see e.g. [5]). This, however, is not the only way. Assuming initial parameter estimates $\mathbf{B}_0$ and $\mathbf{C}_0$ we can minimize the expression in Eqn. (3) by alternating between computing updates $\Delta\mathbf{B}$ that minimize $\|M^{Lin}(\mathbf{B}_0 + \Delta\mathbf{B};\mathbf{C}) - \mathbf{D}\|_2^2$ and updates $\Delta\mathbf{C}$ that minimize $\|M^{Lin}(\mathbf{B}_0;\mathbf{C}_o + \Delta\mathbf{C}) - \mathbf{D}\|_2^2$ [34]. Both equations are linear in their unknowns and

The Alternating Fitting Algorithm

Initialization

Randomly initialize $\mu, \mathbf{B}^1, \mathbf{B}^2, \mathbf{W}, \mathbf{Q}^1, \mathbf{Q}^2; \mathbf{C}_1, \mathbf{C}_2$

Iterate

(I1) Compute $\Delta\Gamma$ in
$\arg\min_{\Delta\Gamma} \|M(\Gamma + \Delta\Gamma; \mathbf{C}_1, \mathbf{C}_2) - \mathbf{D}\|_2^2$
Update $\Gamma \leftarrow \Gamma + \Delta\Gamma$
(I2) Compute $\Delta\mathbf{C}_1$ in
$\arg\min_{\Delta\mathbf{C}_1} \|M(\Gamma; \mathbf{C}_1 + \Delta\mathbf{C}_1, \mathbf{C}_2) - \mathbf{D}\|_2^2$
Update $\mathbf{C}_1 \leftarrow \mathbf{C}_1 + \Delta\mathbf{C}_1$
(I3) Compute $\Delta\Gamma\mathbf{C}_2$ in
$\arg\min_{\Delta\mathbf{C}_2} \|M(\Gamma; \mathbf{C}_1, \mathbf{C}_2 + \Delta\mathbf{C}_2) - \mathbf{D}\|_2^2$
Update $\mathbf{C}_2 \leftarrow \mathbf{C}_2 + \Delta\mathbf{C}_2$

Fig. 9 The alternating fitting algorithm.

The Joint Fitting Algorithm

Initialization

Randomly initialize $\mu, \mathbf{B}^1, \mathbf{B}^2, \mathbf{W}, \mathbf{Q}^1, \mathbf{Q}^2; \mathbf{C}_1, \mathbf{C}_2$

Iterate

(I1) Compute $\Delta = (\Delta\Gamma, \Delta\mathbf{C}_1, \Delta\mathbf{C}_2)$ in
$\arg\min_{\Delta} \|M(\Gamma + \Delta\Gamma; \mathbf{C}_1 + \Delta\mathbf{C}_1, \mathbf{C}_2 + \Delta\mathbf{C}_2) - \mathbf{D}\|_2^2$
Update $\Gamma \leftarrow \Gamma + \Delta\Gamma$
Update $\mathbf{C}_1 \leftarrow \mathbf{C}_1 + \Delta\mathbf{C}_1$
Update $\mathbf{C}_2 \leftarrow \mathbf{C}_2 + \Delta\mathbf{C}_2$

Fig. 10 The joint fitting algorithm.

can therefore be solved directly. In the case of linear models this alternated least squares algorithm has been shown to always converge to the global minimium [2].

PCA does not generalize to bilinear or quadratic models, however the alternating algorithm does. (Note that for bilinear models and fully labeled data, the iterative Tenenbaum-Freeman algorithm can be used [33]). We can minimize Eqn. (2) by solving separately in turn for updates $\Delta\Gamma$, $\Delta\mathbf{C}_1$, and $\Delta\mathbf{C}_2$. See Figure 9. In each case the corresponding minimization problem is linear in its unknowns and can therefore be solved directly. In order to e.g. compute the basis update $\Delta\Gamma$ we compute $\arg\min_{\Delta\Gamma} \|\mathbf{E} - \mathbf{T}_{\Delta\Gamma}\Delta\Gamma\|_2^2$, with the current reconstruction error $\mathbf{E} = \mathbf{D} - M(\Gamma; \mathbf{C}_1, \mathbf{C}_2)$ and the constraint matrix $\mathbf{T}_\Gamma$. $\Delta\Gamma$ can be computed in closed form as $\Delta\Gamma = (\mathbf{T}_\Gamma^T\mathbf{T}_\Gamma)^{-1}\mathbf{T}_\Gamma^T\mathbf{E}$. $\Delta\mathbf{C}_1$ and $\Delta\mathbf{C}_2$ are computed in a similar manner.

While the alternating algorithm works well for linear models it has issues for higher-order models. The linearization into separate component updates ignores the coupling between the bases Γ and coefficients $\mathbf{C}_1, \mathbf{C}_2$. As a consequence the algorithm is more prone to local minima (see results in Section 3.4). To improve performance we propose to *jointly* solve for updates to all parameters at the same time.

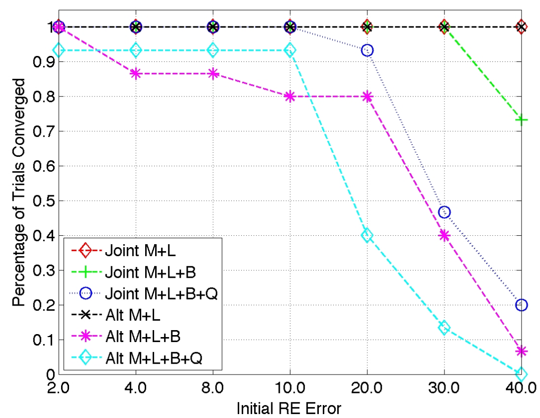


Fig. 11 Comparison of the covergence frequency for the alternating and joint fitting algorithm on synthetic data. The fitting algorithms are initialized with ground-truth data perturbed by noise of varying magnitude. Results are shown for different model configurations combining the mean (M), linear (L), bilinear (B), and quadratic (Q) components. The joint fitting algorithm is more robust as shown by higher frequencies of convergence across models and initial perturbations.

By dropping second order terms and reorganizing components we can transform the minimization problem $\arg\min_{\Delta\Gamma,\Delta\mathbf{C}_1,\Delta\mathbf{C}_2} \|M(\Gamma+\Delta\Gamma;\mathbf{C}_1+\Delta\mathbf{C}_1,\mathbf{C}_2+\Delta\mathbf{C}_2)-\mathbf{D}\|_2^2$ into a similar form as above:

$$\underset{\Delta\Gamma,\Delta\mathbf{C}_1,\Delta\mathbf{C}_2}{\arg\min} \|E-\mathbf{T}_{\Gamma,\mathbf{C}_1,\mathbf{C}_2}\begin{pmatrix}\Delta\Gamma\\ \Delta\mathbf{C}_1\\ \Delta\mathbf{C}_2\end{pmatrix}\|_2^2 \tag{4}$$

with $\mathbf{E}=\mathbf{D}-M(\Gamma;\mathbf{C}_1,\mathbf{C}_2)$ and the constraint matrix $\mathbf{T}_{\Gamma,\mathbf{C}_1,\mathbf{C}_2}$. Figure 10 summarized the algorithm. See [13] for details.

In order to compare the performance of the alternating and joint fitting algorithms we use synthetic data with known ground-truth. We randomly generate bases and coefficient matrices (drawn from a zero mean, unit variance normal distribution) and perturb both with varying amounts of noise before initializing the fitting algorithm. For each noise level the bases and coefficients are then normalized to ensure that all models are initialized at the same reconstruction error. We evaluate the fitting algorithms by comparing the ground-truth models with the fitted models.

In all experiments we report results averaged over five different ground-truth settings with three different initialization settings each for a total of 15 experiments for every model and fitting algorithm. We run every algorithm until convergence (normalized ground-truth error falls below a threshold) or a maximum of 150 iterations, whichever comes first. Figure 11 compares the frequency of convergence for different variations of the joint and alternating fitting algorithms for different initial reconstruction errors. Across all conditions, the joint fitting algorithm performs better than the alternating algorithm. For the combined linear, bilinear and quadratic model (M+L+B+Q) the joint algorithm converges in 80% of all cases whereas the alternating algorithm only converges in 61% of trials. The difference is even larger for the combined linear and bilinear model (M+L+B) where the joint algorithm converges in 96.2% of all trials compared to 68.6% for the alternating algorithm. The joint fitting algorithm also converges faster, requiring on average 8.7 iterations in comparison to 86.7 iterations for the alternating algorithm (for an initial ground-truth error of 20.0).

3.3 Multi-Factor Models with Constraints

The joint fitting algorithm described in Section 3.2 computes bases and coefficients iteratively by minimizing the model reconstruction error for a given training dataset. See Eqn. (2). While the resulting model succeeds at reconstructing the data, no other properties (e.g. affinity of class coefficients, basis orthonormality) are enforced. In order to accomplish this we add further constraints to the energy function on the coefficients, the bases or both. We then strive to compute

$$\begin{aligned} \arg\min_{\Gamma,\mathbf{C}_1,\mathbf{C}_2} \quad & \sum_{l=1}^{n} \|M(\Gamma;\mathbf{c}_1(l),\mathbf{c}_2(l)) - \mathbf{d}_l\|_2^2 + \\ & + \lambda_1 \Theta_1(\mathbf{C}_1,\mathbf{C}_2) + \lambda_2 \Theta_2(\Gamma) \end{aligned} \tag{5}$$

where Θ_1 and Θ_2 refer to sets of constraints. The parameters λ_1 and λ_2 balance the magnitude of the terms.

Let $S^1 = \{s_1^1,\ldots,s_{m_1}^1\}$, $S^2 = \{s_1^2,\ldots,s_{m_2}^2\}$ be sets of coefficient indices of elements in $\mathbf{C}_1$ and $\mathbf{C}_2$, respectively, for which we want to enforce equality. We then strive to compute

$$\begin{aligned} \arg\min_{\Gamma,\mathbf{C}_1,\mathbf{C}_2} \quad & \sum_{l=1}^{n} \|M(\Gamma;\mathbf{c}_1(l),\mathbf{c}_2(l)) - \mathbf{d}_l\|_2^2 + \\ & + \lambda_{11} \sum_{\substack{s_i^1,s_j^1 \in S^1 \\ s_i^1 \neq s_j^1}} \|\mathbf{c}_1(s_i^1) - \mathbf{c}_1(s_j^1)\|_2^2 + \\ & + \lambda_{12} \sum_{\substack{s_i^2,s_j^2 \in S^2 \\ s_i^2 \neq s_j^2}} \|\mathbf{c}_2(s_i^2) - \mathbf{c}_2(s_j^2)\|_2^2 \end{aligned} \tag{6}$$

Linearalizing the expression in Eqn. (6) as described in Section 3.2 leads to

$$\begin{aligned}\underset{\Delta\Gamma,\Delta\mathbf{C}_1,\Delta\mathbf{C}_2}{\arg\min} \quad & \|\mathbf{E}_{RE} - \mathbf{T}_{\Gamma,\mathbf{C}_1,\mathbf{C}_2}\begin{pmatrix}\Delta\Gamma\\ \Delta\mathbf{C}_1\\ \Delta\mathbf{C}_2\end{pmatrix}\|_2^2 + \\ & + \lambda_{11}\|\mathbf{E}_{C_1} - \mathbf{T}_{S_1}\Delta\mathbf{C}_1\|_2^2 \\ & + \lambda_{12}\|\mathbf{E}_{C_2} - \mathbf{T}_{S_2}\Delta\mathbf{C}_2\|_2^2 \end{aligned} \tag{7}$$

with the reconstruction error $\mathbf{E}_{RE} = \mathbf{D} - M(\Gamma;\mathbf{C}_1,\mathbf{C}_2)$, the coefficient constraint error for $\mathbf{C}_1$ ($\mathbf{C}_2$ is defined analogously)

$$\mathbf{E}_{C_1} = \begin{pmatrix}\mathbf{c}_1(s^1_{i_1}) - \mathbf{c}_1(s^1_{i_2})\\ \cdots \\ \mathbf{c}_1(s^1_{i_{m-1}}) - \mathbf{c}_1(s^1_{i_m})\end{pmatrix} \tag{8}$$

and the coefficient constraint matrices $\mathbf{T}_{S_1},\mathbf{T}_{S_2}$. The problem defined in Eqn. (7) can be solved as constraint least squares problem with linear equality constraints (see e.g. [6]). To do so we stack the components of Eqn. (7) and compute

$$\begin{aligned}\underset{\Delta\Gamma,\Delta\mathbf{C}_1,\Delta\mathbf{C}_2}{\arg\min} \| & \begin{pmatrix}\mathbf{E}_{RE}\\ \lambda_{11} * \mathbf{E}_{C_1}\\ \lambda_{12} * \mathbf{E}_{C_2}\end{pmatrix} - \\ - & \begin{pmatrix} & \mathbf{T}_{\Gamma,\mathbf{C}_1,\mathbf{C}_2} & \\ 0 & \lambda_{11} * \mathbf{T}_{S_1} & 0 \\ 0 & 0 & \lambda_{12} * \mathbf{T}_{S_2}\end{pmatrix}\begin{pmatrix}\Delta\Gamma\\ \Delta\mathbf{C}_1\\ \Delta\mathbf{C}_2\end{pmatrix}\|_2^2\end{aligned} \tag{9}$$

The solution to Eqn. (9) can be computed in the same way as the solution to the unconstrained least squares problem. Since the coefficient constraints are added individually and independently for the factors $\mathbf{c}_1$ and $\mathbf{c}_2$, the framework enables semi-supervised learning (see [17]). Constraints on the basis vectors can be enforced in a similar fashion.

3.4 Experiments

In order to evaluate the multi-factor model proposed in Section 3.3 we use a subset of the CMU Multi-PIE face database [15] containing 100 subjects displaying neutral, smile and disgust expressions in frontal pose and with frontal illumination. The images were captured within minutes of each other as part of a multi-camera, multi-flash recording. We normalize the face images by manually establishing facial feature point labels, computing an Active Appearance Model [9,22] over the dataset, and extracting the appearance parameters for all images. We compare privacy protection and data utility of the (ε,k)-map algorithm [13] using two different data representations: the original AAM appearance parameters and the combined $\mathbf{c}_1$ and $\mathbf{c}_2$ parameters extracted from a combined linear and quadratic model. The (ε,k)-map algorithm is a probabilistic extension of the k-Same algorithm described in Section 2.3. For both representations we de-identify the data, reconstruct the (normalized)

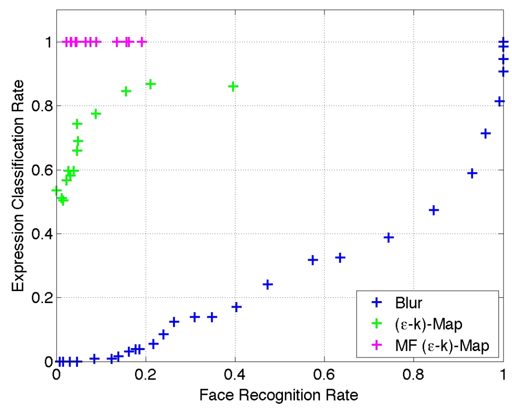


Fig. 12 Privacy-Data Utility map of the (ε, k)-map algorithm using original and multi-factor representations. We show PCA face recognition and SVM facial expression classification rates for different values of the privacy parameter k. Usage of the multi-factor representation (MF (ε, k)-map) results in higher expression classification accuracies than the original representation while providing similar privacy protection. As comparison we also show results for image blurring.

image and compute recognition rates using a whitened cosine distance PCA classifier [4] with the de-identified images as probe and the original images as gallery. We evaluate the utility of the de-identified images by computing facial expression classification rates using a SVM classifier (trained on independent original images) [28]. Figure 12 plots the results of both experiments for varying values of k for the original and multi-factor representations. Across all values of k, expression classification on de-identified images based on the multi-factor representation yields better recognition rates while providing the same privacy protection. As comparison, results for simple blurring of the images are included as well. Figure 13 shows examples of smile images de-identified using the proposed framework.

4 Conclusion and Future Work

In this chapter we provided an overview of face de-identification. We described previously proposed naïve as well as formal de-identification algorithms and illustrated their shortcomings. We then introduced a novel de-identification framework using multi-factor models and demonstrated that the algorithm protects privacy (as measured by face recognition performance) while preserving data utility (as measured by facial expression classification performance on de-identified images).

The multi-factor de-identification algorithm described here operates on single images. However, since it is integrated with the Active Appearance Model framework [9, 22], extension of this work to video de-identification is natural. In Figure 14 we show example frames of applying the algorithm to a video sequence from

Fig. 13 Examples of smile images de-identified using the multi-factor (ε, k)-map algorithm.

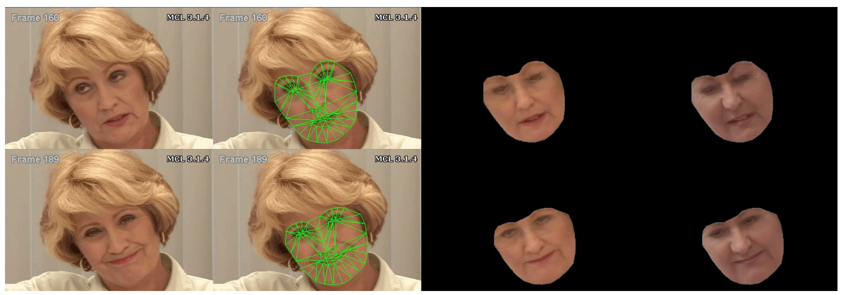


Fig. 14 Comparison of *k*-Same and multi-factor de-identification on video sequences. The multi-factor algorithm preserves more of the data utility during de-identification as shown by e.g. the wrinkles around the eyes of the subject.

the UNBC-McMaster Shoulder Pain Expression Archive [1]. This dataset contains image sequences recorded of subjects after shoulder surgery who rotate either their affected or unaffected shoulder, resulting in a range of pain expressions. We show example images from a sequence in Figure 14, contrasting the results from applying the multi-factor and *k*-Same algorithms. The multi-factor algorithm preserves more of the data utility during de-identification as shown by e.g. the wrinkles around the eyes of the subject.

References

1. A. Ashraf, S. Lucey, J.F. Cohn, T. Chen, Z. Ambadar, K. Prkachin, P. Solomon, and B.-J. Theobald. The painful face - pain expression recognition using active appearance models. In

ICMI, 2007.
2. P. Baldi and K. Hornik. Neural networks and principal component analysis: learning from examples without local minimia. *Neural Networks*, 2(1):53–58, 1989.
3. A. Barucha, C. Atkeson, S. Stevens, D. Chen, H. Wactlar, B. Pollock, and M.A. Dew. Caremedia: Automated video and sensor analysis for geriatric care. In *Annual Meeting of the American Association for Geriatric Psychiatry*, 2006.
4. R. Beveridge, D. Bolme, B.A. Draper, and M. Teixeira. The CSU face identification evaluation system. *Machine Vision and Applications*, 16:128–138, 2005.
5. C.M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1995.
6. A. Björck. *Numerical methods for least squares problems*. SIAM: Society or Industrial and Applied Mathematics, 1996.
7. M. Boyle, C. Edwards, and S. Greenberg. The effects of filtered video on awareness and privacy. In *ACM Conference on Computer Supported Cooperative Work*, pages 1–10, Philadelphia, PA, Dec 2000.
8. Y. Chang, R. Yan, D. Chen, and J. Yang. People identification with limited labels in privacy-protected video. In *International Conference on Multimedia and Expo (ICME)*, 2006.
9. T. Cootes, G. Edwards, and C.J. Taylor. Active appearance models. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 23(6), 2001.
10. J. Crowley, J. Coutaz, and F. Berard. Things that see. *Communications of the ACM*, 43(3):54–64, 2000.
11. F. Dufaux, M. Ouaret, Y. Abdeljaoued, A. Navarro, F. Vergnenegre, and T. Ebrahimi. Privacy enabling technology for video surveillance. In *Proc. SPIE 6250*, 2006.
12. D.A. Fidaleo, H.-A. Nguyen, and M. Trivedi. The networked sensor tapestry (nest): a privacy enhanced software architecture for interactive analysis of data in vide-sensor networks. In *Proceedings of the ACM 2nd international workshop on video surveillance and sensor networks*, 2004.
13. R. Gross. *Face De-Identification using Multi-Factor Active Appearance Models*. PhD thesis, Carnegie Mellon University, 2008.
14. R. Gross, E. Airoldi, B. Malin, and L. Sweeney. Integrating utility into face de-identification. In *Workshop on Privacy Enhancing Technologies (PET)*, June 2005.
15. R. Gross, I. Matthews, J. Cohn, S. Baker, and T. Kanade. The CMU multi-pose, illumination, and expression (Multi-PIE) face database. Technical Report TR-07-08, Carnegie Mellon University, Robotics Institute, 2007.
16. R. Gross, L. Sweeney, F. de la Torre, and S. Baker. Model-based face de-identification. In *IEEE Workshop on Privacy Research in Vision*, 2006.
17. R. Gross, L. Sweeney, T. de la Torre, and S. Baker. Semi-supervised learning of multi-factor models for face de-identification. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.
18. S. Hudson and I. Smith. Techniques for addressing fundamental privacy and disruption tradeoffs in awareness support systems. In *ACM Conference on Computer Supported Cooperative Work*, pages 1–10, Boston, MA, Nov 1996.
19. I.T. Jolliffe. *Principal component analysis*. Springer, second edition, 2002.
20. T. Koshimizu, T. Toriyama, and N. Babaguchi. Factors on the sense of privacy in video surveillance. In *Proceedings of the 3rd ACM workshop on continuous archival and retrival of personal experiences*, pages 35–44, 2006.
21. I. Martinez-Ponte, X. Desurmont, J. Meessen, and J.-F. Delaigle. Robust human face hiding ensuring privacy. In *Workshop on the integration of knowledge, semantics and digital media technology (WIAMIS)*, 2005.
22. I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.
23. C. Neustaedter and S. Greenberg. Balancing privacy and awareness in home media spaces. In *Workshop on Ubicomp Communities: Privacy as Boundary Negotiation*, 2003.
24. C. Neustaedter, S. Greenberg, and M. Boyle. Blur filtration fails to preserve privacy for home-based video conferencing. *ACM Transactions on Computer Human Interactions (TOCHI)*, 2005. in press.

25. E. Newton, L. Sweeney, and B. Malin. Preserving privacy by de-identifying facial images. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):232–243, 2005.
26. P. J. Phillips, H. Moon, S. Rizvi, and P.J Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.
27. H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *International Journal of Computer Vision*, 56(3):151–177, 2002.
28. B. Schoelkopf and A. Smola. *Learning with Kernels: support vector machines, regularization, optimization, and beyond*. MIT Press, 2001.
29. C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998.
30. L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness, and Knowledge-Based Systems*, 10(5):557–570, 2002.
31. S. Tansuriyavong and S-i. Hanaki. Privacy protection by concealing persons in circumstantial video image. In *Proceedings of the 2001 Workshop on Perceptive User Interfaces*, 2001.
32. D. Taubman and M. Marcellin. *JPEG 2000: image compression fundamentals, standards and practice*. Kluwer Academic Publishers, 2002.
33. J.B. Tenenbaum and W. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000.
34. F. de la Torre and M. Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54(1-3):117–142, 2003.
35. K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: principles and practice of background maintenance. In *IEEE International Conference on Computer Vision*, pages 255–261, 1999.
36. M. Vasilescu and D. Terzopoulous. Multilinear subspace analysis of image ensembles. In *Computer Vision and Pattern Recognition*, 2003.
37. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.
38. J. Wickramasuriya, M. Alhazzazi, M. Datt, S. Mehrotra, and N. Venkatasubramanian. Privacy-protecting video surveillance. In *SPIE International Symposium on Electronic Imaging (Real-Time Imaging IX)*, 2005.
39. C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Learning*, 19(7):780–785, 1997.
40. Q. Zhao and J. Stasko. Evaluating image filtering based techniques in media space applications. In *ACM Conference on Computer Supported Cooperative Work*, pages 11–18, Seattle, WA, Nov 1998.