Face De-identification

Song Yuhao

A thesis submitted in partial fulfilment of the requirements
for the degree of
Master of Philosophy

Principal Supervisor: Prof. YUEN Pong Chi
Co-Supervisor: Dr. ZHANG Hui

Hong Kong Baptist University

September 2016

DECLARATION

I hereby declare that this thesis represents my own work which has been done after registration for the degree of MPhil at Hong Kong Baptist University, and has not been previously included in a thesis or dissertation submitted to this or any other institution for a degree, diploma or other qualifications.

Signature: _____

Date: September 2016

SONG Yuhao
Hong Kong Baptist University

Master of Philosophy
September 2016

Face De-identification

Abstract

Acknowledgements

This project would not have been completed without the support of many people. I highly appreciate my supervisor, Hui Zhang, who enlightened me from knowing nothing in this field and helped me make sense of my confusion. Thanks to my principle supervisor, Pong-Chi Yuen, who offered guidance and support during the period of study. Thanks to Hong Kong Baptist University for providing me a monthly studentship for my study. And finally, thanks to my parents and numerous friends who always offering support and love.

Contents

List of Tables

List of Figures

Notation

$\alpha, \beta, \lambda, ..., a, b, c, ...$	Scalars
$(\mathbf{a}, \mathbf{b}, \mathbf{c}, ...)$	Vectors
$(A, B, C, ...)$	Matrices
$(\mathcal{A}, \mathcal{B}, \mathcal{C}, ...)$	Tensors
AAM	Active Appearance Mode
HOSVD	Higher-Order Singular Value Decomposition
SVM	Supported Vector Machine
PCA	Principle Component Analysis
CP Decomposition	CANDECOMP/PARAFAC Decomposition
ALS	Alternating Least Square
ASM	Active Shape Model

Chapter 1

Introduction

Suppose in a social experiment, the designers expect to observe people's reactions when strangers ask for a help to them. The experiment results are recorded as images and videos. To express the research output to public, the recorded data is required to be shown. In this case, the privacy of participants in images or videos must be well protected. Meanwhile, the data should still have the ability to describe the expected research output.

We concentrate on protecting privacy by de-identifying face regions in images and videos. In this chapter, we describe the current research works on face de-identification and the challenges in section 1.1, then give out a summary about our approaches in section 1.2, then list the contributions in section 1.3, at last demonstrate the thesis outline in section 1.4.

1.1 Background

With the development of camera technologies, the image and video acquisition is becoming easier. Nowadays, mounts of applications are centered around image data. Surveillance videos are covering more and more places due to the wide deployment of camera devices. On the other aspect, the advances of computing hardwares and computer vision algorithms make it almost effortless to collect, store and analyze massive image and video data. The best face recognition algorithms, like FaceNet [46], DeepFace [50], can achive more than 95% accuracy in *LFW* and Youtube video datasets. It means that the current machine recognizer perfoms well and stably regardless of the complex background, variant illuminations and unknown face poses. A recent research indicates that it is possible to infer personal informations from a single face image [3]. Therefore, privacy protection rises as an important problem during the sharing of image and video data. With the purpose of protecting privacy, images and videos with people visible in the scene are prohibited to be shared in some applications. For example, in Google Streetview Service which offers high quality street view images, the faces of people in the scene are blurred in the service website currently. Another proper example is a surveillance system monitoring patients in nursing home [1]. The identity of patients in the surveillance video has to be removed before sharing. Since the face is one of the most significant biometric features for a person, our research focuses on protecting the privacy in images and videos through face de-identification, which aims at removing identity information from faces.

Different challenges exist in face de-identification for images and videos. For images, multiple types of information could be extracted from one face image, such as identity, expression, skin color, gender, age, etc. Among of

them, only identity information is related to privacy. The other types of information are defined as data utility of a face image. The key point of face de-identification is to keep the balance between privacy protection and data utility preservation. For instance, the expression information must be preserved in the images from a medical face database aiming at demonstrate the painful faces [6]. For videos, one more point is required in face de-identification. As a set of continuous images, a video is de-identified frame by frame separately. Thus the identity of all frames after processing should not change among adjacent frames. This would disturb the audiences during the video playback. To summarize, there are two challenges in face de-identification for images and videos:

1. Keeping the balance between privacy protection and data utility preservation for images and videos.

2. Keeping the de-identified identity invariant for videos.

To overcome the challenges, plenty of related research works have been released. The most common method is obfuscating images such as pixelization or blurring [10, 4]. Because of the simple implementation, the image obfuscation method is suitable not only to images, but also to videos, such as TV interviews. However, in this approach, the face regions are unreadable to humans after obfuscation. Only replacing the face region with another natural face could preserve the non-privacy related information while removing privacy information. A formal de-identification algorithm, k-same framework, is then proposed [42, 24, 25]. The algorithm takes the average of k closest faces as the de-identified result. As a consequence, the k-same framework is only workable to the person specific database, in which each person has just one image. Other formal algorithms, such as face swapping, face synthetic from

multiple persons, fail to address the problem of presreving data utilities [8, 41]. Except for the obfuscation methods, all the formal approaches are not suitable to face de-identification in videos. Therefore, we try to extent the processing algorithms to larger databases and videos.

1.2 Summary of Proposed Approaches

We use the active appearance model (AAM) [15, 39] and higher order tensors analysis to de-identify the faces in images and videos. The AAM projects a face image into a vector space and represents it with a set of coefficients. The model representation helps avoid ghost faces [25]. Considering the balance between privacy and data utility, we wish to decompose a face image into multiple dimensions so that only the privacy related factors are altered. Therefore, we build up a higher-order tensor and analyze it. Tensor analysis, also known as multilinear algebra, makes the assumption that images are formed as the result of multiple factors. Furthermore, these factors are amenable to linear analysis as each factor is allowed to vary in turn, while the remaining factors are held constant [55, 56].

For images, the face de-identification algorithm is proposed based on the tensor CANDECOMP/PARAFAC(CP) decomposition [34, 32]. We construct a tensor using multiple types of images, then project a new input one into this tensor. After representing the input image by identity and other types of parameters, we can pick out the identity factor and fuse it with other identity parameters from different persons. At last, we de-identify a face image by reconstructing it with altered identity parameters and its untouched parameters. The advantage of our tensor-based algorithm is that the de-identification process could focus only on privacy related information so that the other *data*

utilities could be well preserved.

Among kinds of tensor decomposition algorithms, the CP decomposition estimates the parameters for each dimension using Alternating Least Square (ALS) method. In this way, any face images could be decomposed properly. However, it is not suitable to face de-identification in videos. The reason is that the CP decomposition is sensitive to initial values and would produce different values due to different initial guesses. For videos, AAM is used to represent face images and the Higher-order Singular Value Decomposition(HOSVD) [33] is used to decompose one frame into multiple dimensions. One set of them is privacy related, called the *identity* factors, and the others are non-privacy related factors [22, 35]. With the basis subtensors in HOSVD, this approach produces the same computation results for every computation. During the video de-identification, one frame in the video is picked out and its identity factors are altered to produce a set of de-identified identity factors. For all the other frames in the video, each of them is reconstructed by only replacing the privacy related factors with the de-identified identity ones. Therefore, the identity of de-identified result could keep constant.

1.3 Contributions

In this thesis, we have done some work on face de-identification in images and videos. Our work contributes on three aspects:

1. We use tensor analysis in face de-identification. By decomposing the face regions into privacy related factors and other non-privacy related factors, the proposed algorithms could focus on removing privacy information so that all the other factors are leaved untouched. Therefore, our approach

is suitable to the datasets with multiple factors, such as *expressions, poses, illuminations,* etc. Furthermore, each person in the dataset could have more than one image.

2. In image processing, our algorithm is workable to the images not involving in the tensor. The tensor CP decomposition is used to process images. Since the parameters for each dimension is estimated by initial guess using ALS, the CP decomposition algorithm enlarges the representation ability for face images. We also firstly use rank-n approximation. The increasement of value n helps the face representation more precise, especially for the images never appear in tensors.

3. We succeed to extend the face de-identification algorithms to videos. The existing methods add obfuscations to the regions related to privacy. Compared to the priveous algorithms, the proposed one has two advantages. Firstly, our algorithm produces natural de-identified results from videos. Because the non-privacy related factors are untouched during de-identification, the results could keep the data utility such as *expressions, skin colors,* etc. in a face. Secondly, our method could de-identify a series of images and keep the resulting identity invariant throughout the video playback.

1.4 Thesis Outline

In this thesis, we develope a framework to remove privacy related factors and preserve other data utility factors in images and videos. The rest of this thesis is structured as following. In Chapter 2, we demonstrate the previous research works on face de-identification in images and videos individually.

Chapter 3 introduces foundamental theories of the proposed algorithm: tensor analysis and Active Appearance Model. In Chapter 4 and Chapter 5, the face de-identification in images and videos are explained separately. Chapter 6 expresses the summary of our works and give out conclusions.

Chapter 2

Related Works

Numbers of research works related to face de-identification have been published. For images and videos, the challenges are different. In this chapter, the previous face algorithms for images and videos are described individually in section 2.1 and section 2.2.

2.1 Approaches for Images

For images, the most important point is to keep balance between privacy protection and data utility preservation. For example, a face image shows a person is smiling in frontal pose. After de-identifcation, the person should not be recognized as the original one, but still smile in frontal pose.

The existing algorithms for face de-identification in images falls into three categories: ad-hoc approaches, k-same framework and other face synthesis approaches.

2.1.1 Ad-hoc Approaches

As the most intuitive idea of hiding information, image distortion is widely used to protect privacy. By destorying the original image information, image distortion algorithms make a trade off between privacy information and image quality. General face recognition algorithms find out the identity by computing the similarity between a target image and the images in database. The ad-hoc approaches could protect privacy since the original information has been altered. In traditional broadcast media, such as TV interview and newspapers, the privacy related region is obfuscated to prevent the leak risking of privacy information. This section introduces the ad-hoc approaches including pixelation, blurring, scrambling, masking, cartoon, and encryption.

Pixelation and blurring. The pixelation method subsamples the image by replacing all pixels within a block region with one pixel. In other words, one image is covered by multiple patches [10]. In early computer graphic applications, such as graphic games, the pictures appear at very low resolutions, looking like pixelation. Blurring method smoothes images with some filters like Gaussian filter or average filter [4, 11]. This method alters a pixel value according to its neighbour pixels. As a consequence, the processed image is visually fuzzy. To be more specific, the Gaussian filter is:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}, \tag{2.1}$$

where σ is the standard deviation, x, y are the distances from the origin in the horizontal and vertical axis separately. This equation would produce a weight matrix which would be used to filter the whole image.

Although these two methods are widely used in practical situations like

TV interviews and newspapers, they suffer from the same risk proposed in [42] called *parrot attack*. Blurring and pixelation destory a target image information so that the target image is not similar to any image in database. Parrot attack would preprocess all the database images with the same pixelation or blurring, then applies face recognition algorithms to pixelated or blurred images. Because the target image and the images in database are both processed with similar procedures, the recognition ratio increases sharply.

Scrambling. Dufaux [17] introduced a privacy protection method by scrambling the locations of pixels or Fourier coefficients within a rectangle range. By controling the range size, different degrees of fuzzy results could be produced. It is possible to recover the original image information if the scrambling order is recorded. Essentially, the scrambling method is a kind of encryption with the permutation order as the encryption key. Scrambling method overcomes other traditional encryption methods in original information preserving. If the scrambling range is in proper size, although the result is still unnatural, the original information is understandable. In other words, scrambling also trades privacy protection with data utility. The images shown in Figure 2.1 retriving from [18] illustrate the examples of different de-identification methods introduced in the above content.

Masking. A traditional approach of protection privacy is to cover the region of interest (hereinafter referred to as ROI) with a mask. The mask types are various, many examples are listed in [60]. The intuitive masking method in face de-identificaiton is to cut the face region off and replace the region with black color. It also could be extended to paint the ROI, like face, with background images [57, 44]. After the processing, the ROI is totally ereased as nothing existed.

Cartooning. Converting a real image into cartooning one is a developed

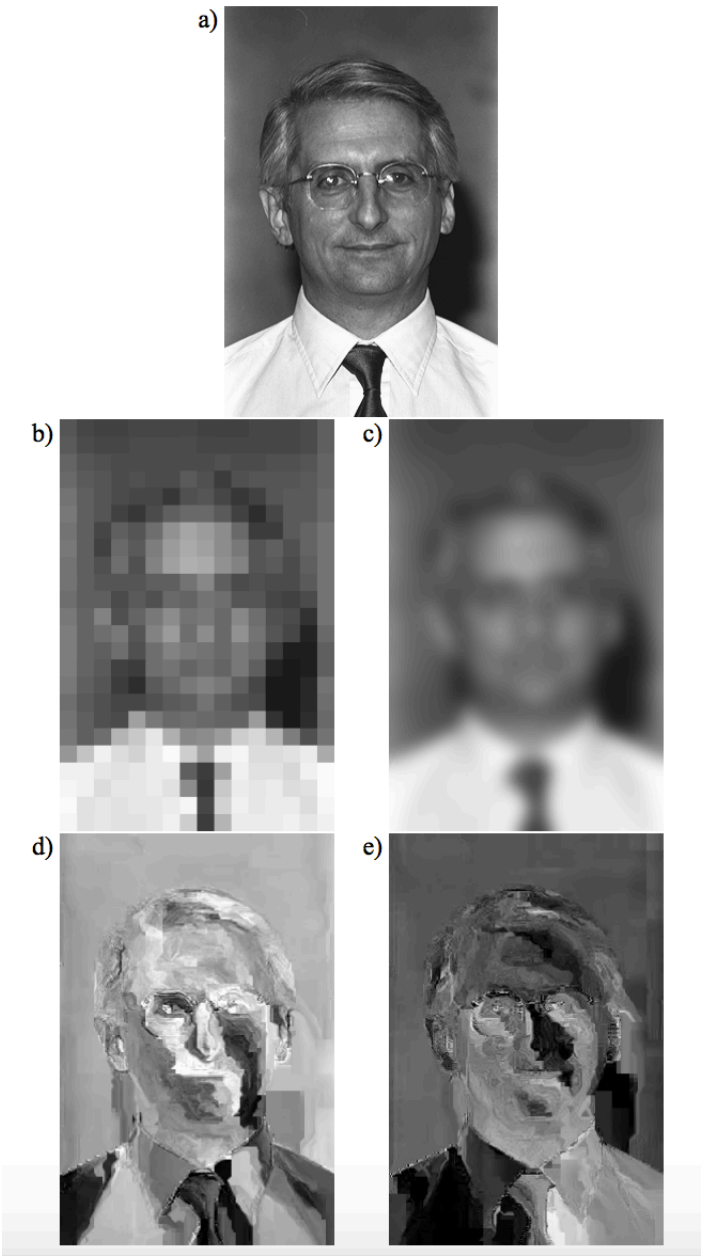

Figure 2.1: Ad-hoc Approaches [18]. (a) Original image. (b) Pixelation. (c) Blurring. (d) Scrambling in pixels. (e) Scrambling in Fourier coefficients.

technology. Erdlyi [21] applied this technology in de-identification area by applying edge detector and mean-shift algorithm in the real image so that the pixels with similar colors would be assign the same value. It is intuitive that cartooning method is able to thwart automatic face recognition algorithms. However, it is hardly deceive a human.

Encryption. Compared with other de-identification methods introduced before, encryption has the advantage in recovery ability [9]. The ROI in image is encrypted to produce snowboard. With a proper key, the encrypted image is always available to be decrypted so that the image remains useful when required. The shortcoming is extra space would be taken to store the encryption key. Furthermore, the processed data is still not visual natural.

As a summation, the previous de-identification methods are all trade-offs between privacy protection and data utility preservation. When processed by the previous introduced methods, the original face image information would be damaged to protect identity but also erase some data utility such as expressions, gender. In another word, the previous introduced methods protect identity privacy information by destorying all the original image information, which definitely destory the data utility simultaneously. It has been a consensus that the key point in face de-identification for images is to keep the balance between privacy protection and data utility preservation. Only replacing a face with another natural face could achive this target.

2.1.2 K-Same Framework

Except for the ad-hoc approaches, this section introduces a formal method which replaces a target face with an average result of k other faces. The approaches belonging to this k-same framework are referred from the k-anonymity

model which is originally used in text information. The k-same approaches introduced here includes: k-same pixel, k-same eigen, k-same model, k-same select and k-same furthest.

2.1.2.1 K-anonymity Model

K-anonymity model is proposed by [49] to solve the problem that a data holder releases a version of private data with scientific guarantees that the individuals involved can not be re-identified while the data remain pratically useful. Suppose one indivual's data is a tuple, this method requires that the released data contain at least k same tuples so that the re-identification ratio for each individual is lower than $\frac{1}{k}$. Originally the model is not specifically designed for face images, the k-anonymity model is applied in various kinds of data. The following table shows the k-anonymity model used in an example of citizen's information from United States.

Race	**Birth**	**Gender**	**ZIP**	**Problem**
Black	1965	m	0214*	short breath
Black	1965	m	0214*	chest pain
Black	1965	f	0213*	hypertension
Black	1965	f	0213*	hypertension
Black	1964	f	0213*	obesity
Black	1964	f	0213*	chest pain
White	1964	m	0213*	chest pain
White	1964	m	0213*	obesity
White	1964	m	0213*	short breath
White	1967	m	0213*	chest pain
White	1967	m	0213*	chest pain

Table 2.1: A K-anonymity example in text information, $k = 2$.

The number k from above example is 2 which means the possibility ratio of

re-identify any individual from the database is less than $\frac{1}{2}$ judging from {Race, Birth, Gender, ZIP}. In other words, at least 2 same tuples in {Race, Birth, Gender, ZIP} would appear in this database. If this is the version of released database, it is possible to get statistic information from this but impossible to match each tuple with specific citizens. Therefore, it is claimed that this approach could keep the balance between privacy protection and data utility preservation.

The theory is then extended to image data. The method used in face de-identification is called k-same framework. The basic idea of this framework is illustrated as the following diagram:

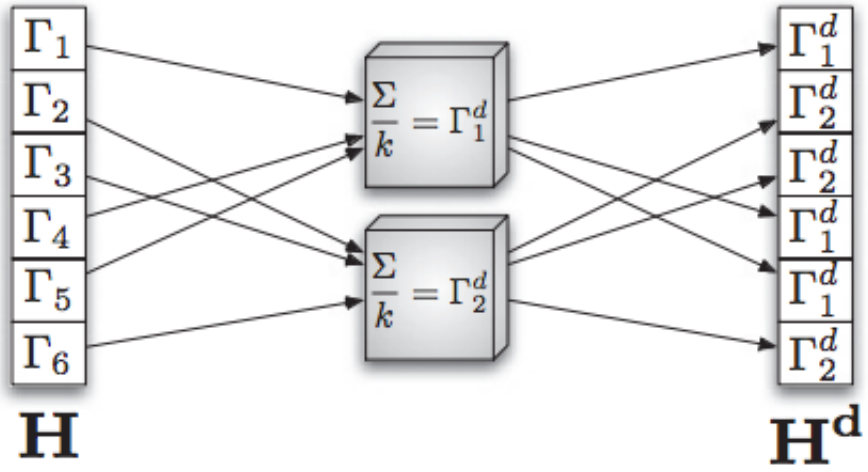

Figure 2.2: Overview of k-same framwork. H is the original image dataset. H^d is the de-identified image dataset.

In k-same framework shown as Figure 2.2, the de-identified result of one image in H^d is the average of its k nearest neighbours from a person specific image set H [26]. In the person-specific image set, each person is represented by just one image.

2.1.2.2 K-Same-Pixel\Eigen

E. Newton firstly applied the K-anonymity theory to privacy protectin by de-identifying face images [42]. Claiming that ad-hoc methods, like pixelation, blurring, are not able to protect privacy, they introduced two new approaches based on K-anonymity model: k-Same-Pixel and k-Same-Eigen. These two algorithms are tested in the person specific database that contains only one image for each person.

For any one face image, A, from the person specific database, the process of k-Same-Pixel is:

- Select k face images that are closest to A including A itself from the person specific database,
- Take the average of these k images as A_{new} by averaging the sum of pixels within face regions,
- Replace the original image A with A_{new},
- Repeat the first step if the recognition ratio of A is higher than $\frac{1}{k}$.

The only difference between k-Same-Pixel and k-Same-Eigen is the later one would take an average image from eigen faces. In mathematics, the eigen face images, composed of eigen values and eigen vectors of original images, are also called projected images, which are computed by Principle Component Analysis (hereinafter referred to as PCA).

Compared to ad-hoc approaches, k-Same-Pixel and k-Same-Eigen are able to combat the parrot attack. It is an improvement for privacy protection. However, the shortcoming is also very obvious. Due to the different size of

multiple faces, the de-identified result produced by these two approaches might be a ghost face. Some examples are illustrated in Figure 2.3.

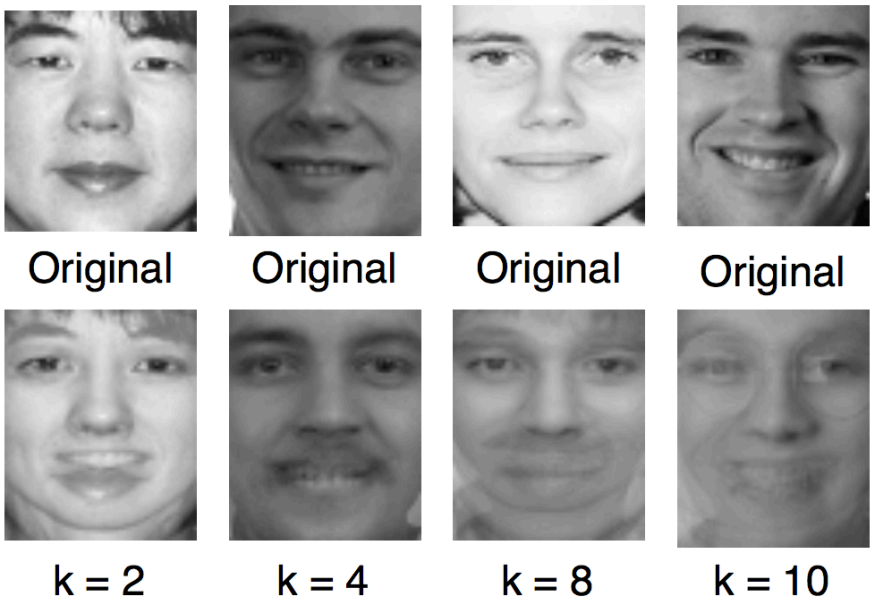

Figure 2.3: Ghost face examples. The upper row are the original images. The lower row are the ghost faces.

Since k faces are stacking together and the face sizes are not the same, the organs in the de-identified face overlaps to form a ghost face. The de-identified results might be able to combat the automatic face recognition algorithms for machines. However, the ghost face is totally unreadable and untolerable subjectively.

2.1.2.3 K-Same-Model

K-Same-Model is designed to solve the ghost face problem [25]. This approach combines a model-based face image parameterization with the k-same framework. For face images, each one is represented as parameters by the active appearance model, in which the shape and appearance of a face

image is represented by parameters separately. The k-same approach is finally applied to these parameters to produce a set of average parameters. At the last step, AAM reconstructs a new face image with the average parameters. To describe the procedures clearly, the pseudo code about k-same-model is described as following:

Algorithm 1: K-Same-Model Algorithm

Input: A person specific dataset H, privacy constant k, $|H| \geq k$, Active Appearance Model $\mathcal{A}$

Output: A de-identified dataset H^d

initialization;

A new dataset $H_o = \varnothing$;

$H^d = \varnothing$;

for $I \in H$ **do**

  Get the AAM parameters representation c_i of I with respect to $\mathcal{A}$;

  Add c_i to H_o;

end

for $c \in H_o$ **do**

  if $|H_o| < 2k$ **then**

    $k = |H_o|$;

  end

  Select k parameters $c_1, ..., c_k$ from H_o that are closest to c according to L_2 norm;

  $avg = \frac{1}{k} \sum_{m=1}^{k} c_m$;

  Add k copies of avg to H_d;

  Remove $c_1, ..., c_k$ from H_o;

end

Since the de-identified processing appears on AAM parameters level, the

result is also a set of parameters of a face image. In AAM, shape and appearance are constructed as two models. The de-identified appearance is finally warped back to de-identified face shape. Therefore, the reconstructed face image by AAM is visual natural.

There is an obvious shortcoming in k-Same-Model. This approach might fabricate information that never exists.

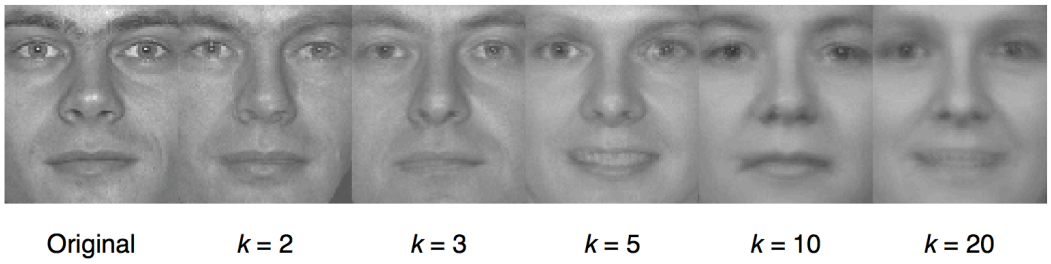

Figure 2.4: Shortcomings of k-same-model. Expressions changes when $k = 5$ and $k = 20$.

As shown in Figure 2.4, the original image is a face with neutral expression. When $k = 5$, the de-identified result by k-Same-Model is a face with smile expression [26]. The reason is that the selected k images have different expressions, so the synthetic face differs from the original one. This example indicates that k-same-model might produce fake information to express different meanings for the original face image.

2.1.2.4 K-Same-Select

The previous introduced approaches have shortcomings in preserving data utility, such as the ability to recognize the original expressions. In order to solve the trade-off problem between privacy protection and data utility preservation in image face de-identification, the k-Same-Select algorithm is introduced [24]. Before taking the average of k images, each face is estimated by a utility

function. For the images with the same utility scores, k images are selected to reconstruct a new face by taking the average. Different with k-Same-Model, this approach introduced in [24] is also appearance-based and works directly on the pixel level. To be more clear, the procedures are described with pseudo code as following:

Algorithm 2: K-Same-Select Algorithm

Input: A person specific dataset H, privacy constant k, $|H| \geq k$, data utility function u

Output: A de-identified dataset H^d

$H^d = \varnothing$;

Let $H_1, H_2, ..., H_l \subset H$ with $H_i = \{x \in H | u(x) = c_i\}$;

$H_i^d = k - same(H_i, k), i = 1, 2, ..., l$;

$H^d = \cap_{i=1}^{l} H_i^d$;

In the algorithm, the utility function is defined according to the data utility types that are required to be preserved. For example, the utility function is to classify expressions if the required data utility is expression. The k-same procedures are applied to the images with the same expression. Therefore, the de-identified result has the same expression with the original image.

2.1.2.5 K-Same-Furthest

For all the approaches belonging to k-same framework introduced before, the kernel procedure is to take average of k closest images to replace the original one so that the image could not be re-identified. The experiments show that k-same approaches work well for the person specific database in which each person has and only has one image. However, the approaches have a common obvious shortcoming. When one person is represented by multiple images, the previous k-same approaches fail to protect privacy. The reason is easy to

understand. The average of k closest images for one person is still that person. Since the de-identified result of the previous k-same approaches is taken from k closest images, the previous k-Same approaches are called k-Same-Closest. Aiming at solving the privacy protection problem in k-Same-Closest, the k-Same-Furthest approach [40, 61] takes the average of k furthest images in L_2 norm to replace the original one. On the other hand, data utility of the result is also the average of the k furthest images. This approach might destory the data utility.

2.1.3 Face Synthesis Approaches

The k-same approaches protect privacy by substituting the original face region with an average of k images. There are many other approaches that substitute the original face region with a synthetic face. This section would introduce some face synthesis approaches used in privacy protection including face swapping [8], face aggregation from donors [41], GARP face [16], face de-identification based on multi-factor model [45] and attribute preserving de-identification method [30].

Face swapping is a popular and simple method to de-identify face images. The most intuitive explanation of this idea is to replace one person's face with another one. For a target face A and another face B, the landmarks such as eye position, nose position is automatically located. Then, all pixels within a face region B is warped to overlap the target face A based on landmarks positions. The research in [8] shows the result from face swapping could be photorealistic. However, the de-identified result contains other's privacy information. This approach would give out soneone's privacy anyway.

Considering of the shortcoming of face swapping, S. Mosaddegh etc. [41]

proposes the method that substitutes the original face with a synthetic face composing of several donors' face parts, such as eyes, nose, mouth. Since the synthetic face contains several persons' characters, it succeeds to deceive machine recognition algorithms and humans without leaking anyone's privacy. Nevertheless, this approach does not consider the problem of data utility preservation.

To balance the privacy protection and data utility preservation, a multi-factor model is used to separate privacy related factors and non-privacy related ones [45]. During the de-identification processing, only the privacy related factors are altered with the non-privacy factors untouched. The multi-factor model in [45] is the combination of linear model, bilinear model and quadratic model. Theoretically, the multi-factor model works when only two data utilities appear in the database. This approach can not guarantee the ability when more types of images are included.

For the image dataset with multiple data utilities, the specific details are processed individually. In [16], gender (G), age (A), race (R) are classified and only the images with the same GAR are blended as the de-identified result. In [30], images are classified according to 17 attributes including gender, race, hair color etc firstly. Then the images with the similar attributes are de-identified. In a summary, the GARP face and attribute preserving algorithm are both similar to k-Same-Select approach.

2.2 Approaches for Videos

Traditional approaches of protecting privacy in videos are to blur or mask the face regions manually which is very cumbersome and time consuming..

Since large amount of video data are producing quickly, automatic face de-identifying method for videos becomes very important. The existing methods follow the same basic structure. That is to detect face regions in all frames and then apply the image obfuscation algorithm. Because of the simplicity, these methods are widely used in practical situations such as TV interviews or surveillance videos [4]. Except for pixelation and blurring, encryption and video scrambling are also general methods to obfuscate face regions[59, 18, 53]. Figure 2.5 shows a encrypted frame from a video.

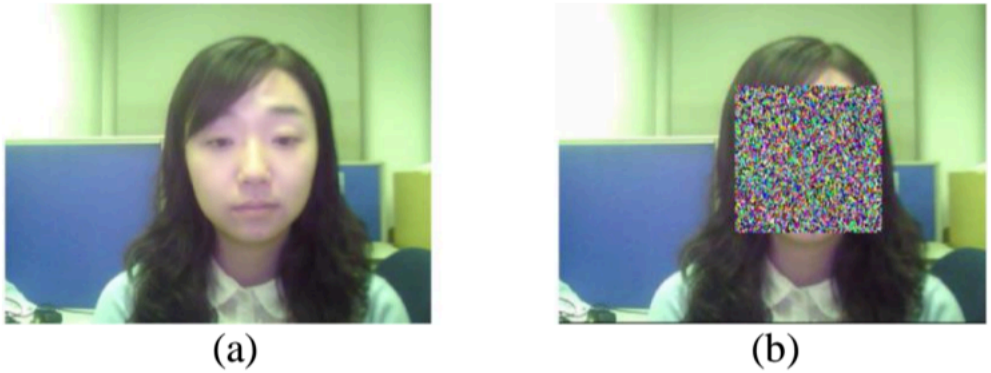

Figure 2.5: One encryption frame. (a) is the original image. (b) is the encrypted image in which face region is covered by snow board.

Inferred from above analysis, only the ad-hoc methods for images are applied to videos. The other algorithms of face de-identification in images are not suitable to process videos, because after de-identification the identity cannot be kept invariant for adjacent images. It would be strange if a person's face changes all the time when a video is playing.

Chapter 3

Foundamental theory

The rest of this thesis would introduce our own approaches of face de-identification for images and videos. Therefore, some foundamental theories need to be explained in advance. In this section, we introduce the concepts of tensor and active apearance model.

To distinguish from scalars, vectors, matrices and higher-order tensors, kinds of symbols are used. Scalars are represented by Greek alphabet letters and low case letters $\alpha, \beta, \lambda, ..., a, b, c, ...$ Vectors are denoted by bold lower case letter $(\mathbf{a}, \mathbf{b}, \mathbf{c}, ...)$. Matrices are shown as $(A, B, C, ...)$. Tensors are described by calligraphic letters $(\mathcal{A}, \mathcal{B}, \mathcal{C}, ...)$. A tensor of order n is denoted by $\mathcal{A} \in R^{i_1 \times i_2 \times ... \times i_n}$, where n is the order. $\mathcal{A}$ can be flattened to a matrix by stacking its mode-m vectors as columns: $A_{(m)} \in R^{i_m \times (i_1 i_2 ... i_{m-1} i_{m+1} ... i_n)}$.

3.1 Tensor

Tensor is the generalized form of scalar, vector and matrix. Formally, tensor is the N-order array. For example, 1-order tensor is vector and 2-order tensor is matrix. The higher-order tensor ($N > 3$) is a multidimensional array. For any N dimensional tensor, it could be unfolded to a matrix. Figure 3.1 shows an unfolding example for a 3-order tensor.

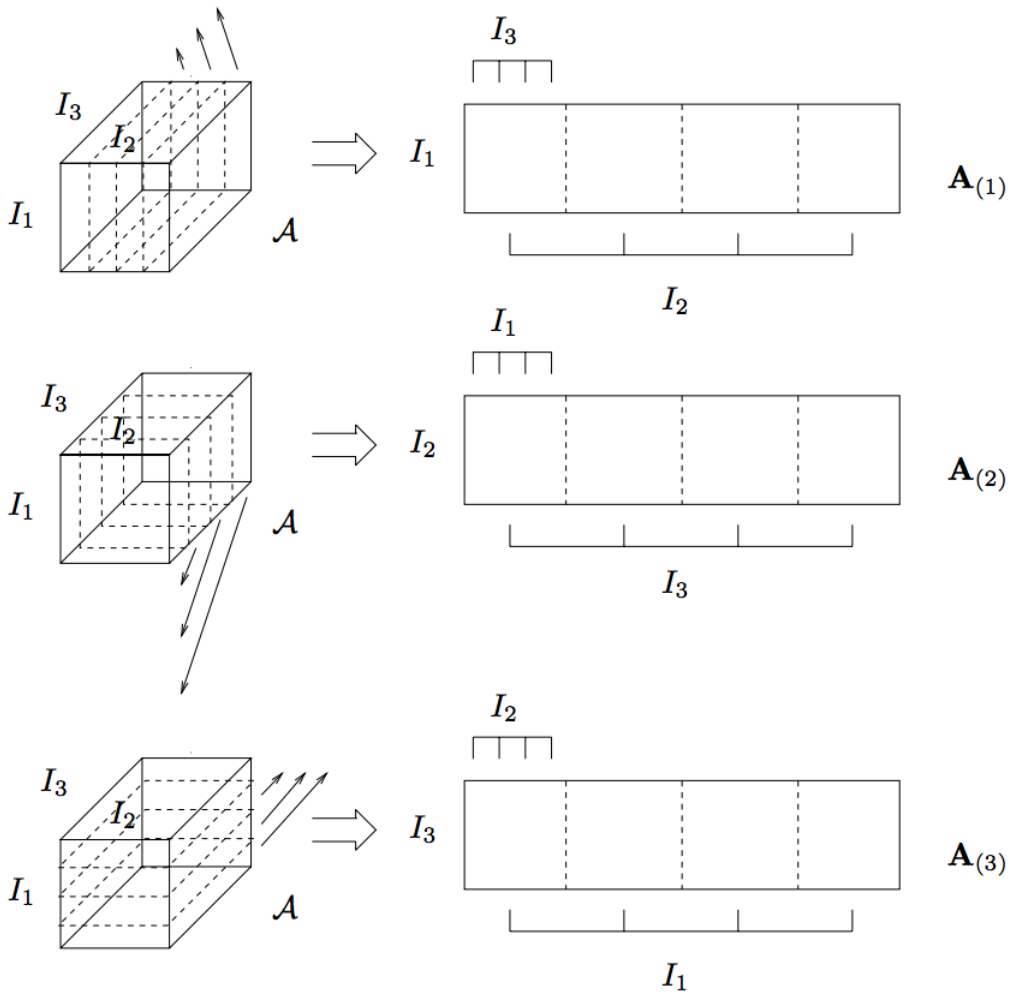

Figure 3.1: Tensor unfolding example. The tensor is $\mathcal{A} \in R^{I_1 \times I_2 \times I_3}$. The unfolded matrixes are $A_1 \in R^{I_1 \times I_2 I_3}$, $A_2 \in R^{I_2 \times I_1 I_3}$ and $A_3 \in R^{I_3 \times I_1 I_2}$.

In mathematical, a $(r+s)$-order tensor is a set of scalars in $(r+s)$ dimensions and it is represented as:

$$\mathcal{A} = a_{i_1 i_2 \ldots i_r}^{j_1 j_2 \ldots j_s}, \tag{3.1}$$

where $a_{i_1 i_2 \ldots i_r}^{j_1 j_2 \ldots j_s}$ is one of the items. We think that a higher-order tensor is an arrangement of data items. We can find the relations between the data items by analyzing the constructed tensor.

3.1.1 Higher-Order Singular Value Decomposition

Generally, an object is indicated as a vector. For example, a 3D point, $\mathbf{a} \in R^3$, is shown as $\mathbf{a} = [x, y, z]$. In face image processing, each item is represented as a vector. With each row as an item, the face images are constructed as a matrix. The PCA could extract the principle component of the data [54, 47], which would find out the most significant common factor among the data. For the data containing two types of variations, a bilinear model could be used to extract the factors individually based on SVD [23, 51, 2]. Similarly, the enough images can be built up as a higher-order tensor. The parameters for each dimension is extracted through HOSVD.

According to the HOSVD [33, 34], any tensor $\mathcal{C} \in R^{i_1 \times i_2 \times \ldots \times i_n}$ could be decomposed as:

$$\mathcal{C} = \mathcal{C}_{core} \times_1 U_1 \times_2 U_2 \ldots \times_n U_n, \tag{3.2}$$

where $\mathcal{C}_{core}$ is a core tensor whose size is the same as $\mathcal{C}$, U_i is the left part in SVD results of corresponding mode-i flattening of $\mathcal{C}$.

3.1.2 CP Decomposition

There are many other tensor decomposition approaches such as CANDELINC, PARAFAC2 [12, 27]. Aiming at decomposing a parameter entity into factors of each dimension, the CP decomposition is applied in this thesis. Firstly introduced in [28, 29], the tensor CP decomposition assumes that any tensor can be represented as a sum of the tensor product by rank-one tensors. [34], shown as (3.3).

$$\mathcal{C} = \sum_{i=1}^{r} \lambda_i * \mathbf{c_1^i} \circ \mathbf{c_2^i} \circ ... \circ \mathbf{c_n^i}, \tag{3.3}$$

where $\circ$ indicates tensor product. It indicates that $\mathcal{C} \in R^{i_1 \times i_2 \times ... \times i_n}$ can be decomposed into r components of tensor product by n vectors.

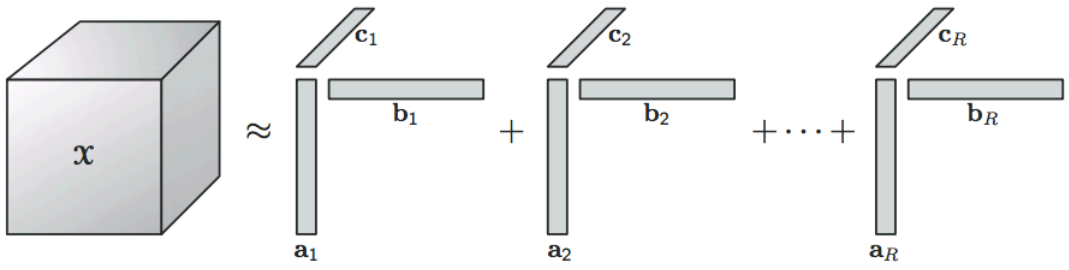

Figure 3.2: The CP decomposition of a 3-order tensor.

Figure 3.2 illustrates the CP decomposition of a 3-order tensor. It is a rank-r decomposition which means the tensor $\mathcal{X}$ is estimated by the sum of r outer product by a_i, b_i, c_i.

3.2 Active Appearance Model

Inteprating a face image with statistical models is a general solution to image representation. The Snake [31] and ASM [14] are used to find face boundary so that the landmarks are located. Considering the appearance, AAM [20, 15, 39] inteprates the face image in two aspects: shape and appearance. The figure 3.3 is the AAM instance for one face image.

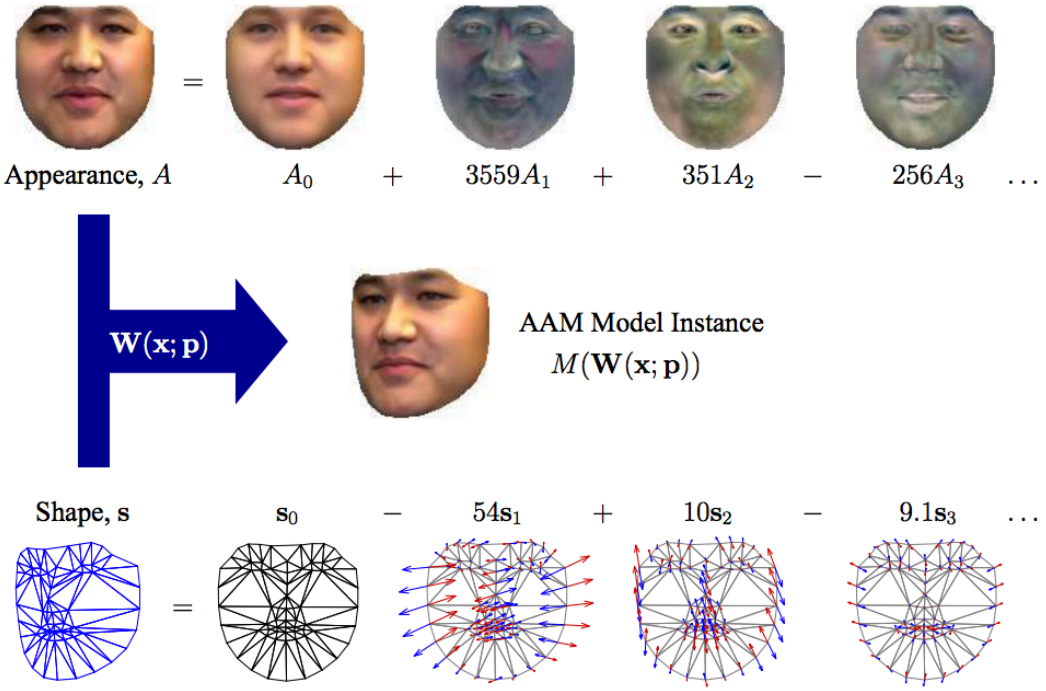

Figure 3.3: An AAM instance. The appearance and shape are deformed individually by altering the coefficients and the appearance is warped to the shape at last.

The shape of a face is represented as the coordinates set of n landmarks $\mathbf{s} = \{x_1, x_2, ..., x_n, y_1, y_2, ..., y_n\}$. The appearance is the set of pixels (the total number is m) within face shape $A = \{p_1, p_2, ..., p_m\}$. In statistical models, all sample images are gathered to get parameters representation.

For shapes,

$$\mathbf{s} = \mathbf{s_0} + \sum_{i=1}^{n} \gamma_i * \mathbf{s_i}, \tag{3.4}$$

where $\mathbf{s_0}$ is the mean appearance of sample images, $\mathbf{s_i}$ is the PCA component of shapes and γ_i is the coefficient for each component.

Similarly, the appearance statistical model is:

$$A = A_0 + \sum_{i=1}^{n} \lambda_i * A_i, \tag{3.5}$$

where A_0 is the mean appearance of sample images, A_i is the PCA component of appearances and λ_i is the corresponding coefficient.

The AAM is widely used in landmarks detection [58], face recognition [19], expression recognition [13], expression transfer [52], etc.

Chapter 4

Face De-Identification in Still Images

With the quick development of Internet and web applications, people are willing to publish self pictures. Furthermore, most of the pictures are allowed to access without permissions. Thus the problem of privacy protection must be considered. This chapter introduces our face de-identification algorithms in still images on tensor CP decomposition.

Tensor analysis, also known as multilinear algebra, makes the assumption that images are formed as a result of multiple factors. Previous researches indicate that it is possible to alter one factor remaining other factors constant[56]. In this chapter, an overview of the algorithm is firstly described and then other procedures are illustrated step by step. At last, related experiments would prove the effevtiveness of the proposed algorithm.

4.1 Algorithm Overview

This section describes the overview of the proposed algorithm. To abstract the method in one word, a face image is decomposed into multiple factors and only the privacy related ones are altered to reconstruct a new face image. This method has advantages in processing the dataset with multiple types of data utilities. The whole procedure of our de-identification algorithm is illustrated in Fig. 4.1.

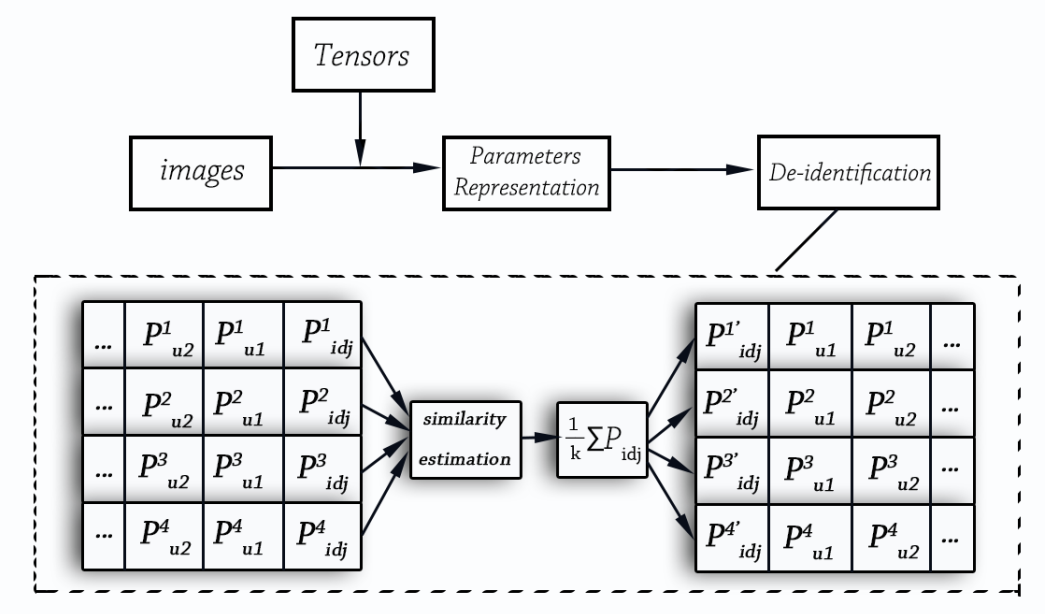

Figure 4.1: Overview of the proposed algorithm. Dashed box is the de-identification process. Each row in the dashed box is one parameters representation of a image. P_{idj} is the *identity* parameters. $P_{u1}, P_{u2}, ...$ are the *utility* parameters.

Firstly, a tensor of face images is pre-constructed. As stated before, we arrange the images with different types of data utilites in some order. The purpose is to extract the common features between these images.

Secondly, we project a face image, img, into the pre-constructed tensor to

get multiple types of parameters. However, we can only get a set of parameters that is a combination of the individual parameters for dimensions. Therefore, we should then decompose the whole parameters into multiple diemensions. One of them is *identity* parameters.

Finally, we select k other images whose *identity* parameters are the farthest from *img* and other types of parameters most closest to *img*. The final de-identified result is the face image which is reconstructed using altered *identity* and other original parameters.

4.2 Tensor Construction

First of all, we pre-construct a tensor of face images according to the required data utilities. It is a container for all image data. A tensor is defined as multiple dimentional array or n-mode array. For example, a vector is 1-mode tensor and a matrix is 2-mode tensor. The n-mode tensor ($n > 2$) is called the higher-order tensor.

According to the theory of PCA, each item of a set of 1-mode data could be represented by the specific parameters and a basis vector. Inferred from that, multiple array has the same property for each dimension. In PCA implementation, each data item is represented by a vector and all data are formed as a matrix at last. Therefore, the images with the same data utility are placed into one dimension in the higher-order tensor so that the common features could be extracted.

To explain more clearly, an example of tensor construction is shown in Figure 4.2. It contains three types of data utility: identities, expressions and poses. Each image is represented as a vector, and the images are placed along

different axis based on the various data utilities. The image tensor with three types of factors is then illustrated in a three dimensional coordinate. When there are more types of the data utility, we could add more dimensions in the coordinate for tensor representation.

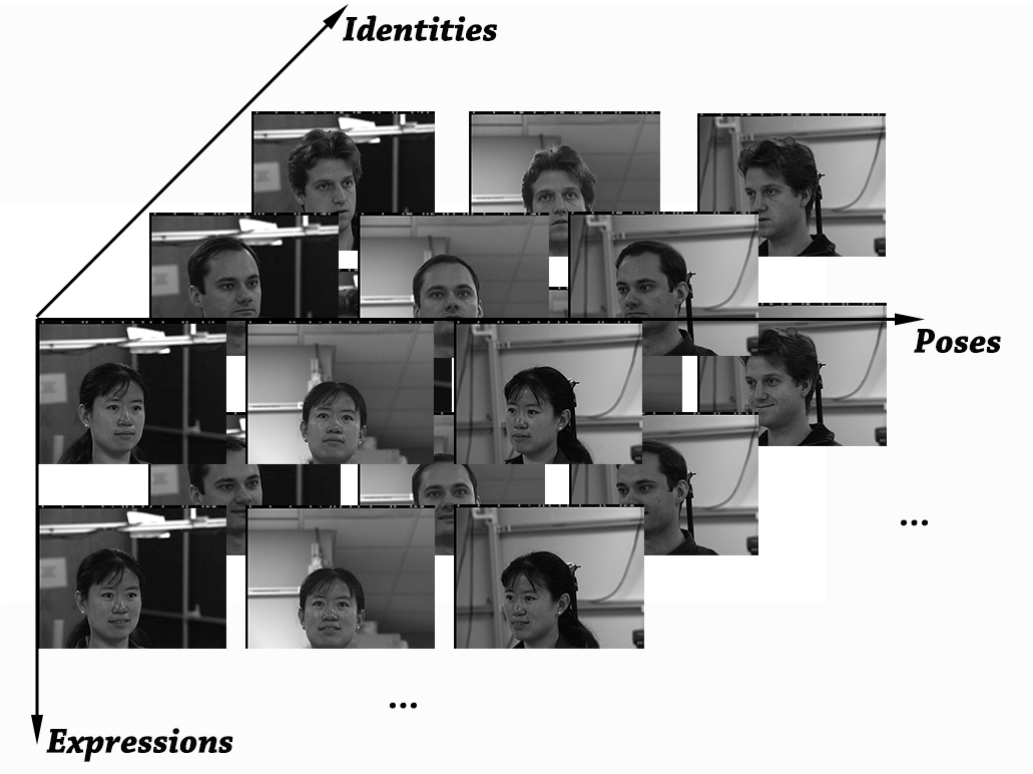

Figure 4.2: Tensor Construction. Each axis is one dimension of the tensor and each dimension in the tensors represents one factor of the images.

Shown as the above figure, all the images along *Poses* axis have the same *identity* and *expression*, but different *poses*. Similarly, the images along *Identities* axis are with the same *pose* and *expression*, but different *identity*. Therefore, the final tensor is a 3-mode array, $\mathcal{A} \in R^{identity \times expression \times pose}$.

4.3 Image Projection

The next step is to project an image into the tensor to get the corresponding parameters. The conventional tensor decomposition theory such as HOSVD [33] is possible to extract parameters of one dimension by defining a basis tensor. However, it is not possible to get the parameters for each dimension. Hence, this section gets a set of parameters for an image and then decomposes it into multiple factors.

As stated before, a 2D face image, img, could be represented by a set of parameters and a tensor. The parameters are a mixture result of multiple factors. Suppose the number of these factors is n, we could construct a mode-(n+1) tensor using proper number of face images, represented as $\mathcal{I} \in R^{i_1 \times i_2 \times ... \times i_n \times i_{data}}$. The i_{data} dimension contains data from face images. For example, if the images are separated into three categories: $identity, expression, pose$, we can build a tensor as $\mathcal{A} \in R^{identity \times expression \times pose \times pixel}$. The pixels are just the data from face images. With this tensor, each img could be represented by a set of parameters.

In the previous researches [56, 35] and many practical examples such as image relighting [36] and expression transfer [38], a tensor is usually decomposed by Higher-Order Singular Value Decomposition (HOSVD) [33]:

$$\mathcal{I} = \mathcal{Z}_{core} \times_1 U_1 \times_2 U_2 ... \times_n U_n \times_{n+1} U_{data}, \tag{4.1}$$

where $\mathcal{Z}_{core}$, called *core tensor*, is the same size as $\mathcal{I}$. One of the matrices in $U_1, U_2, ..., U_n, U_{data}$, U_i is the left part in SVD results of corresponding mode-i flattening of $\mathcal{I}$.

Each *img* in $\mathcal{I}$ could be described as the HOSVD decomposition and a set of vector parameters, $(\mathbf{c_1}, \mathbf{c_2}, ..., \mathbf{c_n})$. Therefore, each *img* is written as:

$$\begin{aligned} img &= \mathcal{Z}_{core} \times_1 (\mathbf{c_1^T} * U_1) ... \times_n (\mathbf{c_n^T} * U_n) \times_{n+1} U_{data} \\ &= (\mathcal{Z}_{core} \times_1 U_1 ... \times_n U_n \times_{n+1} U_{data}) \times_1 \mathbf{c_1^T} ... \times_{\mathbf{n}} \mathbf{c_n^T} \\ &= \mathcal{I} \times_1 \underbrace{\mathbf{c_1^T} \times_2 \mathbf{c_2^T} ... \times_n \mathbf{c_n^T}}_{\mathbf{c_{para}} = \mathbf{c_1} \otimes \mathbf{c_2} ... \otimes \mathbf{c_n}} \\ &= I(data) * \mathbf{c_{para}}, \end{aligned} \tag{4.2}$$

where $I(data)$ is the flattening matrix of $\mathcal{I}$ in mode-$(n+1)$, $\mathbf{c_{para}}$ is the Kronecker product of all kinds of parameters.

Through the above statement, we can naturally conclude that the projection of a face image into the inverse space of a pre-defined image set is the representation parameters for this image in the set. Inferred from the Equ. 4.2, an image *img* is the result of $I(data)$ and $\mathbf{c_{para}}$. Suppose the size of $I(data) \in R^{w*h}$, *img* is unique only when $w \geq h$. Therefore, *K-Nearest Equation Construction* strategy [36] is used to guarantee the constraint. Since $I(data)$ is not promised to be a square matrix, $\mathbf{c_{para}}$ is represented in mathematics as:

$$\begin{aligned} \mathbf{c_{para}} &= I(data)^{-1} * img \\ &= I(data)^T * (I(data) * I(data)^T)^{-1} * img. \end{aligned} \tag{4.3}$$

4.4 Parameters Decomposition

The previous section shows how to project one image into a data space composing of multiple types of images. Each image is represented as a set of parameters. So far, the previous $(\mathbf{c_1}, \mathbf{c_2}, ..., \mathbf{c_n})$ are all assumed to be known,

but we can only get their Kronecker product, $\mathbf{c_{para}}$ which can be seen as the parameters of a face image. Next step is to decompose the entire parameters into different types. It has been tried to get best vector parameters in HOSVD using optimization algorithms [36]. The method is similar to rank-1 approximation of $\mathbf{c_{para}}$ based on tensor CP decomposition, which is used in [35] as *continuous variation estimation*. However, the rank-1 approximation is easily converges to local optimization [34], which would leads to low quality representation. We use rank-r approximation of tensor CP decomposition to decompose the parameters here.

To decompose $\mathbf{c_{para}}$, it is firstly reshaped as a tensor $\mathcal{C} \in R^{i_1 \times i_2 \times ... \times i_n}$ which is the same size as $\mathcal{I}$ in previous n dimensions. According to tensor CP decomposition, any tensor can be represented as a sum of tensor product by vectors [34], shown as (3.3).

$$\mathcal{C} = \sum_{i=1}^{r} \lambda_i * \mathbf{c_1^i} \circ \mathbf{c_2^i} \circ ... \circ \mathbf{c_n^i}, \tag{4.4}$$

where $\circ$ indicates tensor product. It indicates that $\mathcal{C}$ can be decomposed into r components of tensor product by n vectors.

The r value is undetermined. We can pre-define a value *threshold* (e.g. 0.01). The determiend r ($r \geq 1$)should be the first value that makes the deviation between original parameters and reconstructed tensor less than this value. In mathematics, it could be represented as:

$$\| \sum_{i=1}^{r} \lambda_i \mathbf{c_1^i} \circ \mathbf{c_2^i} \circ ... \circ \mathbf{c_n^i} - \mathcal{C} \|_r \leq threshold. \tag{4.5}$$

After the determination of r, one image, *img*, is a explained as a combination of r subimages. Illustrated as (4.6), each of these subimages is formed

as a result of multiple factors.

$$\begin{aligned} img &= \mathcal{I} \times \mathcal{C} \\ &= \mathcal{I} \times (\sum_{i=1}^{r} \lambda_i * \mathbf{c_1^i} \circ \mathbf{c_2^i} \circ ... \circ \mathbf{c_n^i}) \\ &= \sum_{i=1}^{r} \lambda_i * \mathcal{I} \times_1 \mathbf{c_1^i} \times_2 \mathbf{c_2^i} ... \times_n \mathbf{c_n^i}, \end{aligned} \tag{4.6}$$

Compared with the other tensor decomposition methods, the CP decomposition approaches have advantage in processing the images that never appear in the pre-constructed tensor. Using the ALS approximation method, the parameters for each dimension is restored properly. Thus, our method enlarges the representation ability of the pre-constructed tensor.

4.5 De-Identification

It has been shown how to represent a face image by different types of parameters in previous sections. Next step is to de-identify a target with some selected images. The selection method is important. For example, k-same framework fails to protect privacy in a dataset with multiple images for each person because it has to choose the closest neighbors for each image. The decomposition of face images would overcome that problem by selecting images whose *identity* parameters are the farthest from the target and the other *utility* parameters are closest to that target.

As stated in Eq. (4.6), a face image could be represented by r subimages with each one of them being the confluence of multiple factors. One of these factors is *identity*. Since the order is trivial, we assume the $\mathbf{c_1}$ is the *identity* parameters, so $\mathcal{C} = \sum_{i=1}^{r} \lambda_i * \mathbf{c_{idj}^i} \circ \mathbf{c_2^i} \circ ... \circ \mathbf{c_n^i}$. The *identity* parameters are

represented by $\mathbf{c_{idj}}$ in the following text.

Suppose there are two images img_1 and img_2 and their parameters are $\mathbf{c_{img_1}} \in \sum_{\mathbf{i=1}}^{\mathbf{r_1}} \mathbf{c^i_{idj\_1}} \circ \mathbf{c_{2\_1}} ... \circ \mathbf{c_{n\_1}}$ and $\mathbf{c_{img_2}} \in \sum_{\mathbf{i=1}}^{\mathbf{r_2}} \mathbf{c^i_{idj\_2}} \circ \mathbf{c_{2\_2}} ... \circ \mathbf{c_{n\_2}}$. The similarity score between these two images is:

$$\begin{aligned} similarity = \quad & \| \sum_{i=1}^{r_1} \lambda_1^i * \mathbf{c^i_{idj\_1}} - \sum_{i=1}^{r_2} \lambda_2^i * \mathbf{c^i_{idj\_2}} \| - \\ & \sum_{j=2}^{n} \| \sum_{i=1}^{r_1} \lambda_1^i * \mathbf{c^i_{j\_1}} - \sum_{i=1}^{r_2} \lambda_2^i * \mathbf{c^i_{j\_2}} \|. \end{aligned} \tag{4.7}$$

Despite of the complexity of the *similarity* euqation Eq. (4.7), it is summarized as testing the similarity by using the Euclidean distance between images in quantization. Specifically, we use the parameters in *identity* minus the sum of parameters in other *utility* dimensions. For a target image, we can find out the images whose *identity* farthest from it and the other *utility* parameters closest to it by picking out the images with largest *similarity* scores to the target one. Compared to the k-same framework, our approach has the advantage in dividing the parameters into pirvacy related factors and non-privacy related ones. Hence the image classification is more accurate.

For a target image with *identity* parameters $\mathbf{c_{idj\_1}}$, we can find $(k-1)$ images according to (4.7). The *identity* parameters are $\mathbf{c_{idj\_2}}, ..., \mathbf{c_{idj\_k}}$. The de-identified result of the target iamge is:

$$result = I_{(data)} * (\sum_{i=1}^{m} \lambda_i (\frac{1}{k} * \sum_{j=1}^{k} \mathbf{c^i_{idj\_j}}) \circ \mathbf{c^i_2} \circ ... \circ \mathbf{c^i_n}). \tag{4.8}$$

4.6 Approach Summary

Based on a pre-constructed image tensor, a new image is firstly projected to the tensor for parameter representation. Then its parameter is decomposed into *identity* factors and other data utility ones such as *pose*, *expression*. At last, the de-identification process focuses only on *identity* dimension.

The advantage of this approach is to break the limit of person specific dataset. Since the privacy related factors and the non-privacy related ones are separated, our approach picks the *identity* part out and produces a new one according to the *identity* factors from other persons. Therefore, it allows that each person in the dataset could have multiple images. Furthermore, the approach is flexible when more types of images are added dynamically. The pre-constructed tensor could be extended to meet the increasing images.

To make the algorithm clearer, the following steps are listed to summarize the procedures.

Face De-identification in Images Based on Tensor CP Decomposition:

1. Build a tensor according to the required utility types,
2. Project a target image into the tensor to get an entire parameter vector,
3. Decompose the parameter vector into multiple dimensions, one of which is identity,
4. Select k images according to Eq. (7),
5. De-identify the target image according to Eq. (8).

There are still some shortcomings for this approach. The pre-constructed tensor requires a completed dataset. Suppose a 3-mode tensor would be constructed, three types of images are necessary for each item. If three kinds of factors are required, (*pose* (left, frontal, right), *expression* (smile, normal) and *identity*), each person should have 6 images to construct a tensor. It is not easy to get complete dataset. In practical, the missing value is replaced by the average of existed data to solve this problem [22]. That is a useful research problem in the future.

4.7 Experiments and Results

We conduct the experiments on CMU PIE database [48] and IMM database [43]. Sandia tensor toolbox [7] is used to analyze the tensor. The program is executed in Matlab R2014b. In this section, we show the de-identified images and automatic recognition ratios, then explain the reasons why we use rank-r approximation for parameters decomposition.

4.7.1 De-identified Images

For CMU PIE database, a tensor is constructed based on three factors: *identity, pose, expression*. Images from 20 subjects are chosen. Each one of them has 3 poses: frontal, left profile, right profile, and 2 expressions: normal and smile. All selected images are without glasses. The other images are for testing.

Firstly, we use face landmarks detection algorithm from Dlib to get the shapes of all images, then warp the pixels within shape region to a mean shape. Similar as AAM, one image produces two kinds of data, shape and appearance.

The shape remains unchange during de-identification for it influences mainly on *poses* rather than *identity*. Finally, the tensor model is built only on appearance data. In this experiment, appearances are warped into a mean shape to form images as the size of $184 * 194$. It would be time consuming if the appearance is formed as a $(184 * 194) \times 1$ vector. Therefore, 2D appearance representation proposed in [22] is referenced to speed up the computation. Following the procedures described in this chapter, we produce the comparison between original images and de-identified ones as shown in Fig. 4.3.

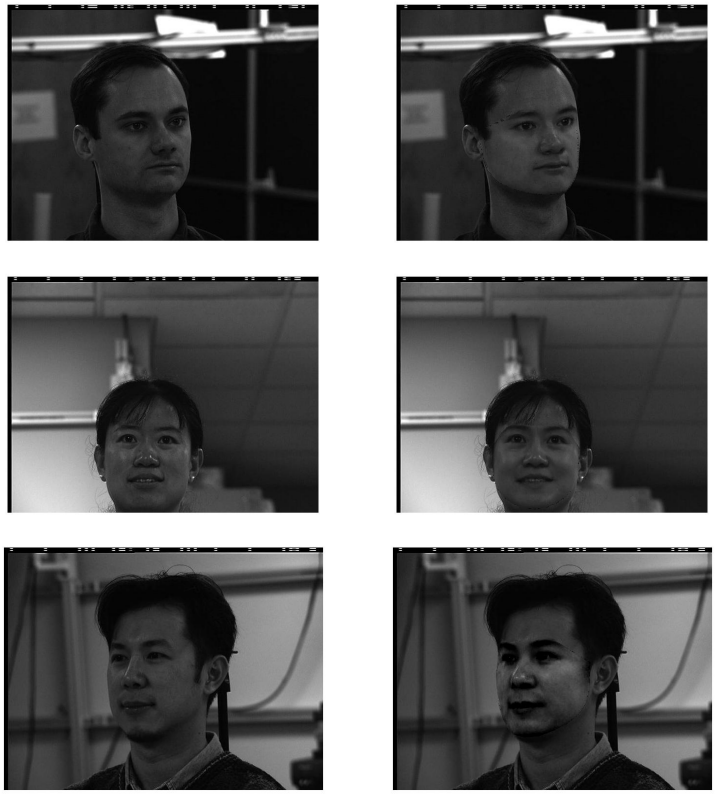

Figure 4.3: Images comparison for various poses and expressions. Left column: original images. Right column: de-identified results.

To show the flexibility of proposed approach, we conduct another experiment on IMM dataset. A tensor is constructed based on two factors: *identity, illumination*. 35 persons are chosen. Each of them has two images under dif-

ferent illuminations. The other images are for testing. With a similar process as in CMU PIE, we get the de-identified images in IMM. Figure 4.4 is one example.

Observing the Fig. 4.3 and Fig. 4.4, the proposed algorithm could produce visual natrual de-identified images and could preserve *data utility*.

Figure 4.4: Images comparison for various illuminations. Left column: original images. Right column: de-identified results.

4.7.2 Face Recognition and Expression Recognition

Except for visual effect, the proposed algorithm should also be verified by automatic recognition algorithms. Using the de-identified results of CMU PIE, we test the quality of privacy protection by face recognition algorithms and test the quality of data utility preservation by expression recognition algorithms.

For identity information, one of the best face recognizers, FaceNet [46] and it opensource implementation [5] are used to test the data before and after de-identification. For expression, a general framework, PCA+SVM, is

used. If the de-identified results could baffle the best face recognizer and can be figured out clearly by a common expression recognizer, the de-identification algorithm is convincible.

State	Face Recognition	Expression Recognition
Before	100%	100%
k = 2	52.50%	100%
k = 3	37.50%	100%
k = 4	31.67%	100%
k = 5	26.67%	100%

Table 4.1: The comparison of face recognition accuracy, expression recognition accuracy before and after de-identification. 'Before' is for original data without de-identification and k is for k images (including the target image) that are used in de-identification.

4.7.3 Reasons of Using Rank-r Approximation

This part explains why we use rank-r approximation rather than rank-1 approximation in parameters decomposition. To represent a face image with multiple vector parameters, rank-1 approximation seems to be the best algorithm. However, rank-1 approximation easily produces low-quality reconstruction images when some features of the input image never appear in the pre-constructed tensor. Rank-r approximation could improve the quality.

As shown in Figure 4.5, an image with glasses causes ridiculous reconstruction result because all the images in pre-constructed tensor are without glasses.

If the features of a image is not included in the pre-constructed tensor, our algorithm is not able to recover it. As an analogy, a 3D point loses one dimension data when projecting it into a 2D plane. We would use rank-n

Figure 4.5: Comparison of reconstruction results by rank-1 and rank-r approximation. Left: original image. Middle: rank-1 approximation. Right: rank-r approximation.

approximation to reconstruct the data as close as possible then de-identify it. Figure 4.6 shows a de-identification example of our model for an image with new features. We use rank-n approximation of a tensor to enlarge the model representation ability.

Figure 4.6: De-identification for an image with new features.

Chapter 5

Face De-Identification in Videos

Different with the previous chapter that focuses on face de-identification about still images, this chapter proposes an approach to de-identify the faces in every frame of a video. Besides the problems mentioned in the previous chapter, one more challenge in face de-identification about videos is to keep the de-identified person constant. Since a person would appear in the adjacent frames of a video, it would be strange if the de-identified person changes between frames.

This chapter would introduce the overview of our proposed algorithm firstly, and then describe the details of our approach in three steps. Section 5.2 demonstrates the build up of the AAM and the higher-order tensor for coefficients. Section 5.3 shows the face de-identification process for a single video frame. Section 5.4 introduces how to extend the face de-identification to other frames in the video. In the following, we summarize the algorithm in section 5.5. Finally, the related experiments are shown to prove the effectiveness of our proposed algorithm.

5.1 Algorithm Overview

Numbers of face de-identification algorithms have been demonstrated in the previous chapter. It indicates that only replacing the original face region with a natural looking face could make the de-identified image readable to humans. With the purpose of keeping balance between privacy protection and data utility preservation as the key problem, the de-identification algorithm for videos also uses face substitution. Compared to the alogrithms for still images, a video is consisted of a set of adjacent frames. It is not workable to simply process the video frame by frame as the still images. Because one person appears in adjacent frames should still be de-identified as the same person.

To achieve this goal, we use AAM and tensor HOSVD to process one frame and then all the other frames with the same way. To make it clear, a flow diagram is illustrated as figure 5.1.

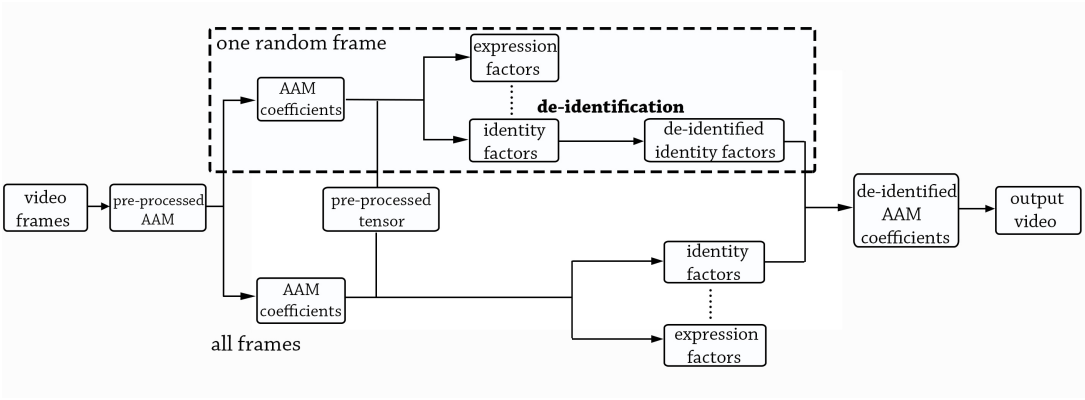

Figure 5.1: Flow diagram of the proposed algorithm. Dashed box is the de-identification process for one frame.

As shown in the diagram, a video is considered as a set of frames. The face region in each frame is represented by a AAM which is constructed by some selected images. These images are from different people and each person

has multiple images that contains different expressions, poses or skin colors. The AAM projects a face image into an image space so that each image is converted into a set of coefficients. It has been proved that a face image could be decomposed into multiple dimensions, such as *identity* factors, *expression* factors, *pose* factors, etc. [56, 35, 22] using tensor analysis. Therefore, we then build up a higher-order tensor based on image coefficients. Based on the theory of HOSVD, a n-mode tensor is obtained from the outer product of a n-mode core tensor and n square matrixes (see details in section 3.1). The AAM coefficients of the image could be decomposed into *identity* factors and other types of non-identity factors by the core tensor and the square matrixes. Finally, during the video de-identification, one of the video frames is picked out randomly. We blend its *identity* factors with k other *identity* factors from different persons to form a set of factors, which we call it as the de-identified *identity* factors. For all the other frames in the video, we also decompose the AAM coefficients into *identity* factors and other ones, then reconstruct the video frames using the de-identified *identity* factors and their own other non-identity factors.

5.2 Pre-processing

In order to get natural de-identified images, we build an AAM for image representation in advance. In this step, an image is projected into a vector space so that it is represented by a set of coefficients [39]. Generally, the model is built up in two levels: shapes and appearances. Since the shape keeps more information about expressions and the appearance is more related to identity information, we focus only on the appearance model. In detail, the pixels within face region are listed as a vector, then a set of images construct

a matrix of the pixels which. Lastly, a vector space is formed according to the PCA results of the pixel matrix.

Following the symbols defined before, the texture of one image is represented as:

$$A = A_0 + \sum_{i=1}^{n} \lambda_i * A_i, \tag{5.1}$$

where A_0 is a mean texture, A_i is a result vector from PCA computation, λ_i is the coefficients of the image.

To decompose the coefficients of an image into *identity* factors and other factors, a higher-order tensor is required. We build up a tensor, called $\mathcal{C}$, of the coefficients according to the types of data utility, such as expressions, poses. Suppose the coefficients of a image $\mathbf{c} \in \mathbf{R^{n \times 1}}$ and two types of data utility need to be preserved: 4 types of expressions and 3 types of poses, the coefficient tensor should be $\mathcal{C} \in R^{4 \times 3 \times n}$. Each dimension of $\mathcal{C}$ subjects to expressions, poses coefficients [56, 22].

According to the HOSVD [33, 32], any tensor $\mathcal{C} \in R^{i_1 \times i_2 \times \ldots \times i_n}$ could be decomposed as:

$$\mathcal{C} = \mathcal{C}_{core} \times_1 U_1 \times_2 U_2 \ldots \times_n U_n, \tag{5.2}$$

where $\mathcal{C}_{core}$ is a core tensor whose size is the same as $\mathcal{C}$, U_i is the left part in SVD results of corresponding mode-i flattening of $\mathcal{C}$.

5.3 Face De-Identification In One Frame

The face de-identification in videos is processed frame by frame. This part shows the process in one frame that is randomly selected. The AAM coefficients of the selected frame would be decomposed and the expected result is a set of de-identified *identity* coefficients.

Suppose we have a set of images containing n_1 identities, n_2 expressions and each of them is represented by a $n_3 \times 1$ vector, the pre-processed tensor is $\mathcal{D} \in R^{n_1 \times n_2 \times n_3}$. A basis tensor about one expression is [35] :

$$\mathcal{D}_{base} = \mathcal{D}_{core} \times_1 U_1 \times_2 U_2(i,:) \times_3 U_3, \tag{5.3}$$

where $\mathcal{D}_{base} \in R^{n_1 \times n_3}$, $U_1 \in R^{n_1 \times n_1}$, $U_3 \in R^{n_3 \times n_3}$, $D_{core} \in R^{n_1 \times n_2 \times n_3}$ and $1 \leq i \leq n_2$. $\mathcal{D}_{base}$ is a basis tensor about the i-th expression in U_2, which means that the product result of $\mathcal{D}_{base}$ and different *identity* coefficients could produce variants images with the i-th expression. Inferred from this example, the basis tensor of a general tensor, $\mathcal{C} \in R^{n_1 \times n_2 \times n_3 \ldots \times n_m}$, is:

$$\mathcal{B}_{base} = \mathcal{C}_{core} \times_1 U_1 \times_2 U_2(i_2,:) \times_3 U_3(i_3,:) \ldots \times_{m-1} U_{m-1}(i_{m-1},:) \times_m U_m, \tag{5.4}$$

where the first dimension of $\mathcal{C}$ is *identity*, the m-th dimension is AAM coefficients and $1 \leq i_j \leq n_j$. Therefore, the AAM coefficients of an image, $img \in R^{n_m \times 1}$, is:

$$img = \mathcal{C}_{core} \times_1 \mathbf{c}_{idj}^T * U_1 \times_2 U_2(i_2,:) \times_3 U_3(i_3,:) \ldots \times_{m-1} U_{m-1}(i_{m-1},:) \times_m U_m, \tag{5.5}$$

where $\mathbf{c}_{idj}$ is the *identity* factors.

With $B \in R^{i_1 \times i_n}$ being the stack matrix of the $\mathcal{B}_{base}$ in its first dimension, the *identity* coefficients of an image could be represented as:

$$\mathbf{c}_{idj}^{T} = B^{-1} * img. \tag{5.6}$$

Therefore, the reconstructed AAM coefficients of this frame is:

$$img_{recon} = B * \mathbf{c}_{idj}^{T}. \tag{5.7}$$

Seen from Equ. 5.4, different values of $(i_2, i_3, ..., i_{m-1})$ produce various reconstructed coefficients. Hence, the basis tensor is determined by choosing the values of $i_2, i_3, ..., i_{m-1}$ that minimize the deviation between reconstructed coefficients and the original ones. It is demonstrated as Equ. 5.8.

$$\underset{i_2,i_3,...,i_{m-1}}{\arg\min} \quad \|img_{recon} - img\|. \tag{5.8}$$

To the present, the *identity* factors, $\mathbf{c}_{idj}$ have been extracted from the whole AAM coefficients. Whether two faces are similar is then measured through Euclidean distance of the *identity* factors:

$$similarity = \quad \|\mathbf{c_{idj\_1}} - \mathbf{c_{idj\_2}}\|. \tag{5.9}$$

For an image, I, we de-identify it by fusing its *identity* coefficients with $(k-1)$ other images whose *similarity* values are the biggest compared to I:

$$\mathbf{c}_{de-idj} = \frac{1}{k} * \sum_{i=1}^{k} \mathbf{c}_{idj}^{i}, \tag{5.10}$$

where $\mathbf{c}_{idj}^{i}$ is the *identity* coefficients of the i-th image. This de-identification method could overcome the shortcomings of k-same framework. The image I is de-identified by the images that are the most different from it rather than the images that are closest to it.

Finally, the de-identified AAM coefficients are computed as Equ. 5.11:

$$\mathbf{v} = \mathcal{B}_{base} \times_1 \mathbf{c}_{de-idj}. \tag{5.11}$$

With the altered AAM coefficients, the new image data, $\mathbf{v}$, is produced. Since the size of $\mathbf{v}$ is the same as the original data, the de-identified image is just the substitution by $\mathbf{v}$.

5.4 Face De-Identification In Videos

A video is a series of frames. The previous section introduces the face de-identification in one frame of a video. As stated before, the face de-identification in videos is processed frame by frame. However, it is not workable to apply the existing methods for image processing to videos directly. The difference is that a person in still images is isolated and a person in videos is continuous for the adjacent frames. The synthetic face is uncontrollable for all the existing face de-identification algorithms.

For each frame in a video, our proposed approach extracts the *identity* factors and other non-privacy related factors. Among of all frames, only one is randomly selected and de-identified to produce the de-identified *identity* coefficients. Since any frame is close to one basis tensors which is described as Equ. 5.4, each frame is de-identified by reconstructing its AAM coefficients

with the de-identified *identity* factors and a basis tensor.

Suppose one frame of a video is selected and its de-identified *identity* factor is $\mathbf{c}_{de-idj}$ and its de-identified AAM coefficient is $\mathbf{v}$, all the frames in this video is reconstructed as:

$$\begin{aligned} A_{recon} &= A_0 + \sum_{i=1}^{n} \mathbf{v}_i * A_i \\ &= A_0 + \sum_{i=1}^{n} (\mathcal{B}_{base} \times_1 \mathbf{c}_{de-idj})_i * A_i, \end{aligned} \tag{5.12}$$

where A_0 and A_i are from Equ. 5.1, $\mathbf{v}$ is the de-identified AAM coefficients from Equ. 5.11 and $\mathbf{v}_i$ is the i-th number. The basis tensor $\mathcal{B}_{base}$ is dynamically determined according to Equ. 5.8 as the frame changes. For example, when the image is in frontal pose and normal expression, we use the corresponding basis tensor.

After all the frames are processed, the de-identified video is just to restore the frames to where they belong as in the original video.

5.5 Algorithm Summary

The previous sections have introduced our implementation of face de-identification process in videos. Similar with the procedures in still images, the most important point of face de-identification in videos is also to keep the balance between privacy protection and data utility preservation. Besides that, one more challenge in videos is to keep the de-identified person constant in all the frames. Normal image face de-identification algorithms such as k-same

framework could not achieve that. Our solution is to separate one frame into privacy related factors and other factors, then altered the privacy related ones only and de-identify all the other frames with the altered privacy factors at last. Theoretically, the proposed algorithm could protect face privacy, preserve the data utility and keep the person constant in all frames after de-identification.

To summarize the procedures of face de-identification in videos, the proposed algorithm is described step by step as:

1. Pre-compute an AAM for a set of selected images;
2. Build a higher-order tensor for the AAM coefficients of all images;
3. Select one frame from a video randomly and get the de-identified *identity* factors, $\mathbf{c}_{de-idj}$;
4. De-identify all frames from the video with $\mathbf{c}_{de-idj}$.
5. Group all the de-identified frames as the de-identified video.

Except for the advantages, the proposed algorithm still has some shortcomings. Since we build up a tensor and decompose it using HOSVD, all images are classified according to the dimensions of the pre-constructed tensor. Therefore, the algorithm is weak when the new types of images appear. Suppose two types of expressions, *normal* and *smile*, exist in tensor, the transmission frame between *normal* and *smile* might be classified as two equally. Mistaken de-identified frames might appear during these transmission frames.

In still image processing, we use tensor CP decomposition to enlarge the representation ability of the pre-constructed tensor. However, it is not workable for videos. Because the CP decomposition estiamte parameters for each dimension using ALS, the result is sensitive to the initial values. It is hard to

get a constant *identity* parameters. In the future, we might add some weight coefficients to the HOSVD result so that the transimission frames could be well represented.

5.6 Experiments and Results

We verify the proposed algorithm through an experiment in the Extended Cohn-Kanade face database(CK+) [37]. The CK+ includes 593 sequences from 123 subjects. Each sequence is a series of frames for one subject to perform one kind of expression. Sandia tensor toolbox [7] is used to analyze the tensor. The program is executed in Matlab R2014b. The experiment would show the de-identified images and then show that the de-identified identity is constant in quantity.

Although the landmarks of all images are well recorded, only parts of the sequences have expression labels. Therefore, we select the frontal images of 4 expressions (neutral, anger, happy, surprise) from 23 subjects to build up an AAM. Since only the texture model is considered in AAM, each image is represented by a set of coefficients, $\mathbf{c} \in R^{80\times 1}$, in AAM. As a consequence, a tensor $\mathcal{C} \in R^{23\times 4\times 80}$ about AAM coefficients is constructed. The images are then processed as stated in Section 5.2 Some de-identified results are shown as following:

Our pre-processed tensor did not involve the factors about skin colors. Therefore, in order to make the de-identified images more natural, the final result adds the value of the difference between the mean of original textures and the mean of de-identified results. Two more results on neutral and happy expressions are illustrated in Fig. 5.3. As the results demonstrated, we can

Figure 5.2: $k = 2$. Each column is a series of frames from a video. The first column is the series of original frames with surprise expression, and the third column is the original ones with anger expression. The second and fourth columns are the de-identified frames respectively.

claim that our algorithm is able to de-identify a series of frames in a video with natural results and keep the de-identified identity invariant. Meanwhile, the proposed algorithm could fully protect the original data utilities, such as expressions.

We can quantify the similarity between the original frames and the de-identified results using the comparison demo, which is supplied by OpenFace [46, 5] and outputs the predicted similarity score of two faces by computing the squared L2 distance between their model representations.

The left subject shown in Fig. 5.2 is taken for example. Eleven original frames and the corresponding de-identified frames are picked out, and one de-identified frame is randomly chosen as a target frame. The similarity between

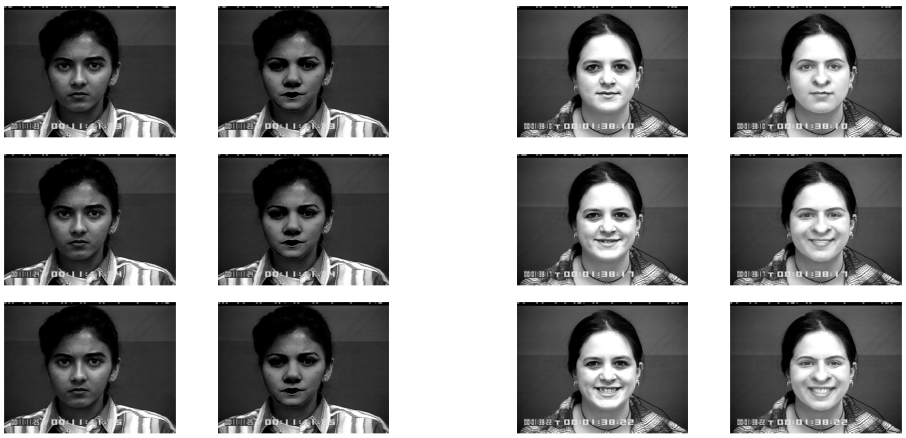

Figure 5.3: De-identified images of different subjects with neutral and happy expressions. $k = 5$. The first and third columns are the original frames. The second and fourth columns are the de-identified resutls respectively.

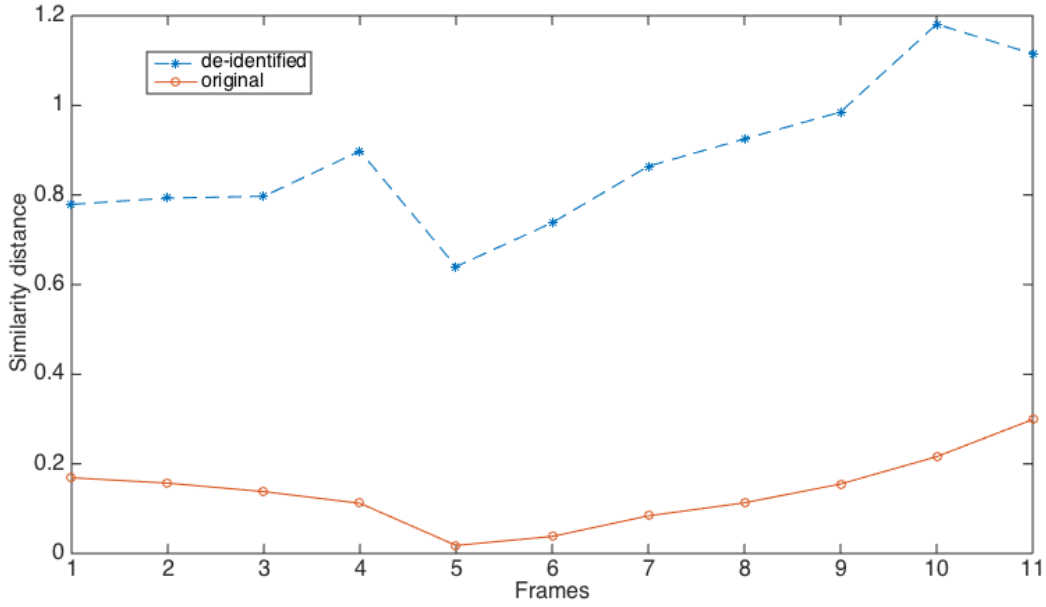

Figure 5.4: Similarity distance. Red solid line: similarity distance between de-identified frames and the target frame. Blue dash line: similarity distance between original frames and the target frame.

the original & de-identified frames and the target frame is shown in Fig. 5.4. Seen from the figure, the de-identified frames are obvious more close to the target frame. Furthermore, the distances between de-identified results are small.

Chapter 6

Conclusion and Future Works

In this thesis, we focus on the privacy protection problem for still images and videos by de-identifying face regions. To make the de-identification algorithm more robust to various images, multi-linear algebra is used to store the data. Hence, the algorithm is more practical.

For the still imags, we propose a novel algorithm of face de-identification based on tensor CP decomposition. The algorithm helps to decompose a face image into multiple dimensions so that the de-identification process could happen only on privacy related information. All the experiments are conducted in the dataset which include multiple images for each person. On the other aspect, the proposed algorithm could process the dataset with multiple factors such as *pose, expressions*. Theoretically, more factors could be appended to the algorithm. Futhermore, the rank-R approximation enlarges the model representation ability for images with new features. One shortcoming is that the proposed algorithm requires a complete dataset to construct a tensor. The complete dataset is not easy to collect for some cases. Therefore, the future work might be the research on de-identification based on tensors with missing

values so that the tensor model could break the limitation of incomplete image data.

For videos, our proposed algorithm use AAM and tensor HOSVD decomposition to de-identify a video by replacing the original faces with other natural ones. Shown as the experiments, we can keep the identity of de-identified frames in a video invariant and protect its data utility, such as expressions simultaneously. There are shortcomings in the proposed method. The transformation frames between expressions, such as from neutral to surprise, creates unnatural results sometimes. This problem might be solved by apply a weight average process to basis selection. This work could be improved by using Tensor-AAM. More data utilities could be added to the model.

Bibliography

[1] A. Barucha, C. Atkeson, S.S.D.C.H.W.B.P., Dew, M.: Caremedia: Automated video and sensor analysis for geriatric care. Annual Meeting of the American Association for Geriatric Psychiatry (2006)

[2] Abboud, B., Davoine, F.: Appearance factorization for facial expression analysis. Bmvc (2004)

[3] Acquisti, A., Gross, R., Stutzman, F.: Face recognition and privacy in the age of augmented reality. Journal of Privacy and Confidentiality 6(2) (2014)

[4] Agrawal, P., Narayanan, P.J.: Person de-identification in videos. In: in Proc. ACCV, 2009. pp. 266–276

[5] Amos, B., Ludwiczuk, B., Satyanarayanan, M.: Openface: A general-purpose face recognition library with mobile applications. Tech. rep., CMU-CS-16-118, CMU School of Computer Science (2016)

[6] Ashraf, A.B., Lucey, S., Cohn, J.F., Chen, T., Ambadar, Z., Prkachin, K.M., Solomon, P.E.: The painful face - pain expression recognition using active appearance models. Image Vision Comput. 27(12), 1788–1796 (2009)

[7] Bader, B.W., Kolda, T.G., et al.: Matlab tensor toolbox version 2.6. Tech. rep., http://www.sandia.gov/ tgkolda/TensorToolbox/ (2015)

[8] Bitouk, D., Kumar, N., Dhillon, S., Belhumeur, P., Nayar, S.K.: Face swapping: Automatically replacing faces in photographs. In: In ACM Transactions on Graphics (Proceedings of SIGGRAPH (2008)

[9] Boult, T.E.: Pico: Privacy through invertible cryptographic obscuration. In: Proceedings of the Computer Vision for Interactive and Intelligent Environment. pp. 27–38. CVIIE '05, IEEE Computer Society, Washington, DC, USA (2005)

[10] Boyle, M., Edwards, C., Greenberg, S.: The effects of filtered video on awareness and privacy. In: Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work. pp. 1–10. CSCW '00, ACM, New York, NY, USA (2000), http://doi.acm.org/10.1145/358916.358935

[11] C. Neustaedter, S.G., Boyle, M.: Blur filtration fails to preserve privacy for homebased video conferencing. ACM Transactions on Computer Human Interactions(TOHI) (2005)

[12] Carroll, J.D., Pruzansky, S., Kruskal, J.B.: Candelinc: A general approach to multidimensional analysis of many-way arrays with linear constraints on parameters. Psychometrika 45(1), 3–24 (1980)

[13] Cheon, Y., Kim, D.: Naturalfacialexpressionrecognitionusingdifferential-aamandmanifoldlearning. Pattern Recognition 42(7), 13401350 (2009)

[14] Cootes, T., Taylor, C., Cooper, D., Graham, J.: Active shape models-their training and application. Computer Vision and Image Understanding 61(1), 38 – 59 (1995), http://www.sciencedirect.com/science/article/pii/S1077314285710041

[15] Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. IEEE Trans. Pattern Anal. Mach. Intell. 23(6), 681–685 (Jun 2001), http://dx.doi.org/10.1109/34.927467

[16] Du, L., Yi, M., Blasch, E., Ling, H.: Garp-face: Balancing privacy protection and utility preservation in face de-identification. In: IEEE International Joint Conference on Biometrics, Clearwater, IJCB 2014, FL, USA, September 29 - October 2, 2014. pp. 1–8 (2014), http://dx.doi.org/10.1109/BTAS.2014.6996249

[17] Dufaux, F., Ebrahimi, T.: Scrambling for Privacy Protection in Video Surveillance Systems. IEEE Trans. on Circuits and Systems for Video Technology vol. 18(no. 8), 1168–1174 (2008)

[18] Dufaux, F., Ebrahimi, T.: A framework for the validation of privacy protection solutions in video surveillance. IEEE International Conference on Multimedia and Expo (ICME) (2010)

[19] Edwards, G.J., Cootes, T.F., Taylor, C.J.: Face recognition using active appearance models. Springer Berlin Heidelberg (1998)

[20] Edwards, G.J., Taylor, C.J., Cootes, T.F.: Interpreting face images using active appearance models. In: IEEE International Conference on Automatic Face and Gesture Recognition, 1998. Proceedings. pp. 145–149 (1998)

[21] Erdelyi, A., Barat, T., Valet, P., Winkler, T., Rinner, B.: Adaptive cartooning for privacy protection in camera networks. In: AVSS'14. pp. 44–49 (2014)

[22] Feng, Z.H., Kittler, J., Christmas, W.J., Wu, X., Pfeiffer, S.: Automatic face annotation by multilinear aam with missing values. In: ICPR. pp. 2586–2589. IEEE Computer Society (2012)

[23] Freeman, W.T., Tenenbaum, J.B.: Learning bilinear models for two-factor problems in vision. In: Conference on Computer Vision and Pattern Recognition. pp. 554–560 (1997)

[24] Gross, R., Airoldi, E., Malin, B., Sweeney, L.: Integrating utility into face de-identification. In: Privacy Enhancing Technologies, 5th International Workshop, PET 2005, Cavtat, Croatia, May 30-June 1, 2005, Revised Selected Papers. pp. 227–242 (2005)

[25] Gross, R., Sweeney, L., Cohn, J.F., la Torre, F.D., Baker, S.: Model-based de-identification of facial images. In: AMIA 2008, American Medical Informatics Association Annual Symposium, Washington, DC, USA, November 8-12, 2008 (2008)

[26] Gross, R., S.L.C.J.d.l.T.F.B.S.: Face de-identification (2009)

[27] Harshman, R.A.: PARAFAC2: Mathematical and technical notes. UCLA Working Papers in Phonetics 22, 30–44 (1972b)

[28] Hitchcock, F.L.: The expression of a tensor or a polyadic as a sum of products. Journal of Mathematics and Physics 6(1), 164189 (1927)

[29] Hitchcock, F.L.: Multiple invariants and generalized rank of a p-way matrix or tensor. Journal of Mathematics and Physics 7 (1927)

[30] Jourabloo, A., Yin, X., Liu, X.: Attribute preserved face de-identification. In: ICB. pp. 278–285. IEEE (2015)

[31] Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision 1(4), 321–331 (1988)

[32] Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. 51(3), 455–500 (Aug 2009), http://dx.doi.org/10.1137/07070111X

[33] Lathauwer, L.D., Moor, B.D., Vandewalle, J.: A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. 21(4), 1253–1278 (Mar 2000), http://dx.doi.org/10.1137/S0895479896305696

[34] Lathauwer, L.D., Moor, B.D., Vandewalle, J.: On the best rank-1 and rank-(r1, r2,...,rn) approximation of higher-order tensor. SIAM Journal on Matrix Analysis and Applications pp. 21–1324 (2000)

[35] Lee, H.S., Kim, D.: Tensor-based aam with continuous variation estimation: Application to variation-robust face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(6), 1–1 (2009)

[36] Lin, D., Xu, Y., Tang, X., Yan, S.: Tensor-based factor decomposition for relighting. In: Proceedings of the 2005 International Conference on Image Processing, ICIP 2005, Genoa, Italy, September 11-14, 2005. pp. 386–389 (2005), http://dx.doi.org/10.1109/ICIP.2005.1530073

[37] Lucey, P., C.J.F.K.T.S.J.A.Z., Matthews, I.: The extended cohn-kanade dataset (ck+): A complete expression dataset for action unit and emotion-specified expression. the Third International Workshop on CVPR for Human Communicative Behavior Analysis pp. 94–101 (2010)

[38] Macedo, I., Brazil, E.V., Velho, L.: Expression transfer between photographs through multilinear aam's. Brazilian Symposium on Computer Graphics and Image Processing pp. 239–246 (2006)

[39] Matthews, I., Baker, S.: Active appearance models revisited. International Journal of Computer Vision 60(2), 135 – 164 (November 2004)

[40] Meng, L., Sun, Z.: Face de-identification with perfect privacy protection. In: International Convention on Information and Communication Technology, Electronics and Microelectronics. pp. 1234–1239 (2014)

[41] Mosaddegh, S., Simon, L., Jurie, F.: Photorealistic face de-identification by aggregating donors' face components. In: Computer Vision - ACCV 2014 - 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part III. pp. 159–174 (2014)

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

[43] Nordstrøm, M.M., Larsen, M., Sierakowski, J., Stegmann, M.B.: The IMM face database - an annotated dataset of 240 face images. Tech. rep., Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby (may 2004), http://www2.imm.dtu.dk/pubdb/p.php?3160

[44] Qureshi, F.Z.: Object-video streams for preserving privacy in video surveillance. In: Tubaro, S., Dugelay, J.L. (eds.) AVSS. pp. 442–447. IEEE Computer Society (2009)

[45] Ralph Gross, Latanya Sweeney, F.d.l.T., Baker, S.: Semi-supervised learning of multi-factor models for face de-identification. In: IEEE Conference on Computer Vision and Pattern Recognition (June 2008)

[46] Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2015)

[47] Shen, W., Khanna, R.: Prolog to face recognition: Eigenface, elastic matching, and neural nets. Proceedings of the IEEE 85(9), 1422–1422 (1997)

[48] Sim, T., Baker, S., Bsat, M.: The cmu pose, illumination, and expression (pie) database of human faces. Tech. Rep. CMU-RI-TR-01-02, Robotics Institute, Pittsburgh, PA (January 2001)

[49] Sweeney, L.: K-anonymity: A model for protecting privacy. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 10(5), 557–570 (Oct 2002), http://dx.doi.org/10.1142/S0218488502001648

[50] Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2014)

[51] Tenenbaum, J.B., Freeman, W.T.: Separating style and content with bilinear models. Neural Computation 12(6), 1247–1283 (2000)

[52] Theobald, B.J., Matthews, I.A., Cohn, J.F., Boker, S.M.: Real-time expression cloning using appearance models. In: Proceedings of the 9th International Conference on Multimodal Interfaces. pp. 134–139. ICMI '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1322192.1322217

[53] Tong, L., Dai, F., Zhang, Y., Li, J., Zhang, D.: Compressive sensing based video scrambling for privacy protection. In: Visual Communications and Image Processing (VCIP), 2011 IEEE. pp. 1–4 (2011)

[54] Turk, M., Pentland, A.: Eigenfaces for recognition. J. Cognitive Neuroscience 3(1), 71–86 (Jan 1991), http://dx.doi.org/10.1162/jocn.1991.3.1.71

[55] Vasilescu, M.A.O., Terzopoulos, D.: Multilinear analysis of image ensembles: Tensorfaces. In: Proceedings of the 7th European Conference on Computer Vision-Part I. pp. 447–460. ECCV '02, Springer-Verlag, London, UK, UK (2002), http://dl.acm.org/citation.cfm?id=645315.649173

[56] Vasilescu, M.A.O., Terzopoulos, D.: Multilinear subspace analysis of image ensembles. In: CVPR (2). pp. 93–99. IEEE Computer Society (2003)

[57] Venkatesh, M.V., ching Samson Cheung, S., Zhao, J.: Efficient object-based video inpainting. Pattern Recognition Letters 30(2), 168 – 179 (2009), video-based Object and Event Analysis

[58] Wang, N., Gao, X., Tao, D., Li, X.: Facial feature point detection: A comprehensive survey. Eprint Arxiv (2014)

[59] Winkler, T., Rinner, B.: Trustcam: Security and privacy-protection for an embedded smart camera based on trusted computing. p. 593600 (2010)

[60] Winkler, T., Rinner, B.: Security and privacy protection in visual sensor networks: A survey. ACM Comput. Surv. 47(1), 2:1–2:42 (May 2014), http://doi.acm.org/10.1145/2545883

[61] Zongji Sun, L.M., Ariyaeeinia, A.: In: Workshop on De-Identification for Privacy Protection in Multimedia (DEID) (2015)