IET Signal Processing

Special Section: De-identification

Efficient approach to de-identifying faces in videos

ISSN 1751-9675
Received on 31st December 2016
Revised 12th August 2017
Accepted on 12th September 2017
E-First on 27th October 2017
doi: 10.1049/iet-spr.2016.0761
www.ietdl.org

Li Meng[1] ✉, Zongji Sun[1], Odette Tejada Collado[1]
[1]School of Engineering and Technology, University of Hertfordshire, College Lane, Hatfield, UK
✉ E-mail: l.1.meng@herts.ac.uk

Abstract: This study presents a novel approach that extends face de-identification from person-specific (closed) sets of facial images to open sets of video frames. Inspired by the previous work in facial expression transfer, the authors have introduced an 'identity shift' to ensure identity consistency within a de-identified video sequence. The 'identity shift' is derived from the first video frame of a person and is then applied in the de-identification of all subsequent frames of the same person. Experimental results show that video frames that are originally associated with the same person will remain related to a common new identity after the application of the proposed approach. In addition, the proposed approach is able to achieve privacy protection as well as preservation of dynamic facial expressions. Finally, MATLAB implementation of the approach has confirmed its potential to operate in real time at the highest standard video frame rate.

1 Introduction

To date, the most successful face de-identification methods are the solutions in the *k*-Same family, where privacy protection is achieved by implementing the theory of *k*-anonymity [1]. All existing *k*-Same methods [2–5] have been proposed for the de-identification of a set of face images, where the set is first divided into clusters of size *k* and then each face image is de-identified by an aggregation (usually the centroid) of its own cluster. Since all *k* faces in each cluster are de-identified with the same aggregated face, these methods have hence been given the name '*k*-Same'. The privacy protection performance of a face de-identification method is evaluated in terms of the re-identification risk of its de-identified faces, i.e. the risk of its de-identified faces being matched with their original identities or face images. The lower the re-identification risk, the better the privacy protection a face de-identification method provides. As each *k*-Same de-identified face image appears in the de-identified image set *k* times and it can be matched, at best, with only one of its *k* original faces, all *k*-Same methods can guarantee a re-identification risk lower than $1/k$ for their de-identified faces. This guaranteed privacy protection level has been further reduced to zero by the *k*-Same-furthest method when being tested against similarity-based face recognition methods [5]. An extension of *k*-Same-furthest is the *k*-Diff-furthest method [6]. Apart from better privacy protection, another main contribution of *k*-Diff-furthest is that it produces a distinguishable de-identified face for each original face. Experimental results in [6] showed that *k*-Diff-furthest is able to maintain diversity among the de-identified faces and keep them as diverse and distinguishable as their original faces. Our work in this paper extends the *k*-Diff-furthest method in order to achieve face de-identification in videos.

Like all existing *k*-Same methods, the *k*-Diff-furthest method takes a set of face images as the input and de-identifies the whole set in a single pass. This means that neither the *k*-Same methods nor the *k*-Diff-furthest method can be applied directly to de-identify a video sequence frame by frame in real time. Furthermore, all these face de-identification methods demand the input image set to be person specific [2], i.e. each person present in the image set has only one image in the set and no two images in the set relate to the same person. This means that we cannot apply the *k*-Same methods or the *k*-Diff-furthest method to a set of frames taken from the same video sequence as they often contain face images of the same person(s). Therefore, a new approach must be developed for the de-identification of faces in a video sequence.

In addition to (i) privacy protection and (ii) preservation of data utility (e.g. gender, age and facial expression), face de-identification in videos presents two more challenges: (iii) identity consistency and (iv) real-time processing. Here, identity consistency means that the face instances originally related to the same identity must relate solely to the same new identity after de-identification. This paper presents a novel approach to de-identifying faces in videos, which is able to address all four challenges. This new approach is inspired by our previous work on data utility preservation through facial expression transfer (FET) [7] and is based on our previously developed *k*-Diff-furthest method [6].

This paper is structured as follows: Section 2 provides a review of FET. Section 3 explains two methods adopted in our new approach – the facial landmark annotation method and the *k*-Diff-furthest method. Section 4 describes our proposed approach to de-identifying faces in videos and Section 5 tests its abilities to address the four challenges. Finally, the findings of this work are summarised in Section 6.

2 Subject review of facial expression transfer

Although *k*-Same methods are able to guarantee *k*-anonymised privacy protection, they seldom address the other common challenge of face de-identification – the preservation of data utility. The *k*-Diff-furthest method shares the same drawback. Since the *k*-Same methods as well as the *k*-Diff-furthest method calculate each de-identified face as the average of *k* faces, the data utility will be averaged out and lost unless all the *k* faces share the same data utility including both the class (e.g. happy or sad) and the intensity of the utility (e.g. how happy a face is). To address the first half of the data utility challenge, the *k*-Same-Select method [4] attempted to integrate utility preservation into face de-identification with a data utility classifier. The utility classifier is used to partition the original face set into mutually exclusive utility subsets. The *k*-Same de-identification is then performed on each utility subset. However, this additional classification step has made the algorithm inflexible as the utility classifier would have to be re-trained every time a new utility class is introduced to the task. Nevertheless, the main drawback of the *k*-Same-Select method is that it preserves only one aspect of each data utility – the class of the utility. For binary data utility such as gender, this is adequate. However, for utilities such as facial expressions, it is necessary to preserve not

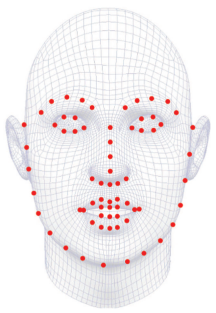

Fig. 1 *MultiPIE/IBUG 68 point facial landmarks (figure taken from [20])*

only the class label (e.g. happy or sad) but also the intensity of the utility (e.g. how happy a face is).

Among the data utilities associated with images/videos of human faces, facial expression is the most complicated utility due to the variety of expressions, the simultaneous appearance of multiple expressions, and the various degrees of intensity with each expression. On the other hand, the preservation of facial expression is of significant value to face de-identification in videos, where various expressions with dynamic degrees of intensity are expected.

In recent years, active appearance model (AAM) [8] has been broadly used for building non-rigid deformable models. In face biometrics, this model provides a compact statistical representation of the shape and the texture variation of human faces. AAM-derived face representations have been employed in [9], where the experimental results proved AAM face representations to be highly effective for the task of facial expression analysis. Subsequently, there have been various attempts to use AAM in FET. The study in [10] focused on real-time dynamic facial expression transfer using AAMs, generating realistic talking faces in real time at low computational cost. The work assessed how a fitted expression from one AAM could be used to synthesise the same expression realistically onto another person or an animated character in a separate AAM. The procedure is able to produce video sequences that are smooth and seemingly acceptable. The study in [11] describes techniques for mapping and manipulating facial gestures and global head movements in video sequences of people engaged in conversation. Such techniques operate in real time at video frame rate due to the simple mapping of parameters between AAMs, without a requirement of high-level semantic information about the facial expressions. Similarly, the work in [12, 13] presented ad-hoc control of the facial expressions of a target actor by cloning the facial expressions from another actor in a source video. Again, the whole process of this approach can be carried out in real time.

Inspired by the effectiveness and, in particular, the efficiency of FET approaches in transferring facial expressions between subjects, an FET approach was previously proposed and integrated into the face de-identification process to recover the facial expressions of an original face on its de-identified face [7]. This FET approach can be represented mathematically as follows:

$$\boldsymbol{\Lambda}_{\rm d}^{\rm expr} = \boldsymbol{\Lambda}^{\rm expr} - \boldsymbol{\Lambda}^{\rm neutral} + \boldsymbol{\Lambda}_{\rm d}^{\rm neutral}, \tag{1}$$

where vector $\boldsymbol{\Lambda}$ denotes the AAM features of a face.

Equation (1) achieves FET in two simple steps: first, calculate the change of AAM features caused by the expression of the original expressive face in comparison to the neutral face of the same person, i.e. $\boldsymbol{\Lambda}^{\rm expr} - \boldsymbol{\Lambda}^{\rm neutral}$, and then apply the same change to the de-identified neutral face $\boldsymbol{\Lambda}_{\rm d}^{\rm neutral}$.

Despite the fact that the FET approach defined in (1) does not satisfy *k*-anonymity, experimental results in [7] showed that it has hardly any impact on the re-identification risk of the de-identified faces generated by the *k*-Same-furthest method. In addition, the FET process enables the expressions of an original face to be effectively cloned onto its de-identified face. As confirmed by the experimental results from data utility evaluation in [7], the proposed FET approach preserves expression better than *k*-Same-Select. Furthermore, visual results of the output faces in [7] demonstrate that FET is able to preserve not only the category but also the dynamic details of facial expressions. As mentioned, our approach to de-identifying faces in videos is based on this FET process.

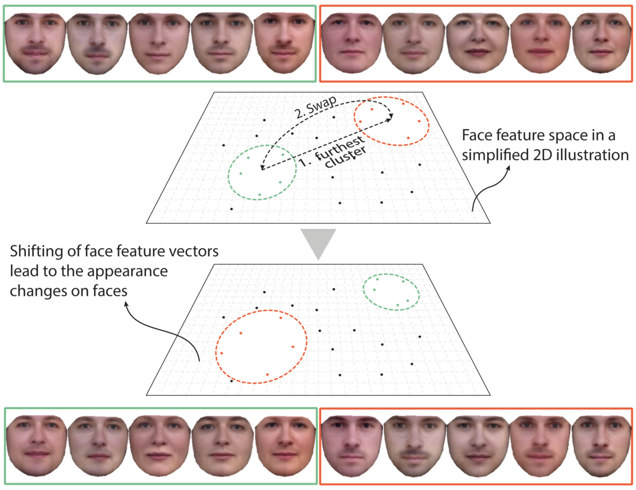
2. Swap
1. furthest cluster
Face feature space in a simplified 2D illustration
Shifting of face feature vectors lead to the appearance changes on faces

Fig. 2 *k-Diff-furthest face de-identification swaps original faces between a pair of clusters in order to retain the diversity of the original face set in the de-identified face set*

3 Supporting methods

3.1 Landmark annotation using constrained local neural field (CLNF)

Facial landmark annotation is an essential first step in face recognition, face de-identification and FET. The accuracy of this first step has a significant impact on the performance of a face biometric system.

There have been many attempts to accomplish accurate and person independent facial landmark annotation [8, 14–17]. One of the most promising is the constrained local model (CLM) method [14]. However, CLM struggles in poor lighting conditions, in the presence of occlusion and when facing unseen datasets. In our system, we make use of CLNF. This extension of CLM incorporates local neural field patch experts (also called local detectors) to exploit spatial relationships between pixels and evaluate the probability of a landmark being aligned at a particular pixel location [18].

Fig. 1 shows the 68 facial landmarks used in this work to define the shape of a face. This set of facial landmarks has been widely used in face biometric systems [3, 19–21].

3.2 The k-Diff-furthest method for face de-identification

The *k*-Diff-furthest method has been proposed to generate unique and distinguishable de-identified faces for a given person-specific set of face images. Instead of using aggregated information to de-identify faces such as in the *k*-Same methods, *k*-Diff-furthest swaps original faces between a pair of clusters in order to retain the diversity of the original face set in the de-identified face set. Fig. 2 depicts this cluster swapping process of the *k*-Diff-furthest method.

In each iteration, the method randomly selects a seed image from the set of remaining original images and simultaneously forms two clusters based on the seed image. One of the clusters $\boldsymbol{C}_{\rm c}$ consists of images closest to the seed and the other cluster $\boldsymbol{C}_{\rm f}$ consists of images furthest to the seed. The formation of each cluster pair of $\boldsymbol{C}_{\rm c}$ and $\boldsymbol{C}_{\rm f}$ must satisfy the condition of no overlap between the two clusters. As already mathematically proved in [6], as long as this condition stands, for each de-identified face there exists at least one original face in the opposite cluster that is closer to it than its original face. In other words, the de-identified faces produced by the *k*-Diff-furthest method have a re-identification risk of zero when there is no overlap between $\boldsymbol{C}_{\rm c}$ and $\boldsymbol{C}_{\rm f}$ and when the

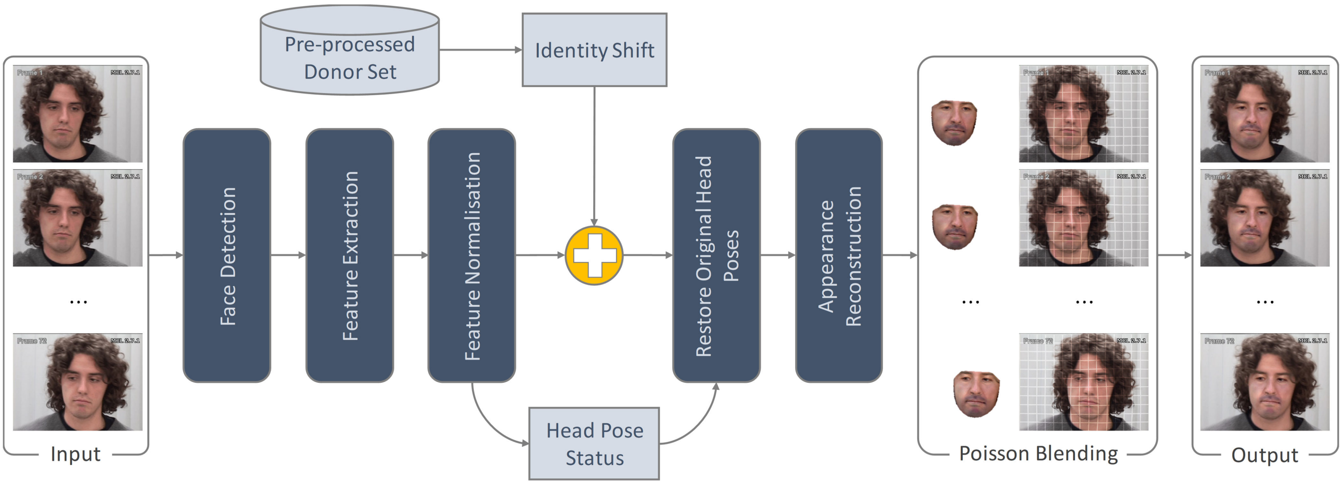

Fig. 3 *Proposed system for de-identifying faces in videos*

de-identified faces are tested against similarity-based face recognition methods. The re-identification risk would average $1/N$ against random matching, where N is the number of identities in the original data set. The privacy protection ability of the k-Diff-furthest method has been tested with the FERET dataset using several face recognition benchmark methods [22]. Although these face recognition methods use different face representation models and hence the condition of no overlap between $\boldsymbol{C}_\mathrm{c}$ and $\boldsymbol{C}_\mathrm{f}$ cannot always be kept in the respective model spaces, the re-identification risk of the k-Diff-furthest de-identified faces has always remained <0.5% for all the face recognition software tested [including PCA, local binary patterns (LBP), HOG and LPQ] [22].

4 Proposed system

4.1 Efficient approach to de-identifying faces in videos

Let $\boldsymbol{\Lambda}^i \in \boldsymbol{P}$ be the face instances in a set of video frames that are related to the same person p. When the FET process defined in (1) is applied to the individual face instances $\boldsymbol{\Lambda}^i$, (1) can be re-written as

$$\boldsymbol{\Lambda}_\mathrm{d}^i = \boldsymbol{\Lambda}^i + (\boldsymbol{\Lambda}_\mathrm{d}^\mathrm{neutral} - \boldsymbol{\Lambda}^\mathrm{neutral}), \tag{2}$$

where vector $\boldsymbol{\Lambda}^\mathrm{neutral}$ is the original neutral face of person p and vector $\boldsymbol{\Lambda}_\mathrm{d}^\mathrm{neutral}$ is person p's de-identified neutral face. We refer to the term $(\boldsymbol{\Lambda}_\mathrm{d}^\mathrm{neutral} - \boldsymbol{\Lambda}^\mathrm{neutral})$ as the *identity shift* of person p as it defines the changes of person p's neutral face when it shifts from its original identity to a new identity.

As stated, all $\boldsymbol{\Lambda}^i \in \boldsymbol{P}$ relate to the same person and hence have the same $\boldsymbol{\Lambda}^\mathrm{neutral}$. Obviously, for identity consistency in the de-identified videos all $\boldsymbol{\Lambda}^i \in \boldsymbol{P}$ must have an identical $\boldsymbol{\Lambda}_\mathrm{d}^\mathrm{neutral}$. This means that in order to de-identify face instances of person p, we just need to calculate the identity shift of person p once. The de-identification of person p's face instances in a video can be achieved frame by frame, highly efficiently, through a simple addition of the input face instance and the calculated identity shift. In addition, our face de-identification approach defined by (2) is able to achieve both privacy protection and preservation of facial expressions simultaneously. The first term of (2) transfers the facial expression from an original face instance onto its de-identified version, while the second term shifts the face instance to its new identity. Furthermore, our approach inherently ensures identity consistency after face de-identification.

Fig. 3 shows the proposed face de-identification system with an example video input. Apart from the key operations defined in (2), additional operations have been employed in the system to achieve better visual quality with the de-identified video. The rest of this section describes the functional blocks in Fig. 3.

4.2 Calculation of identity shift

The identity shift defines the difference between a person's original neutral face $\boldsymbol{\Lambda}^\mathrm{neutral}$ and its de-identified version $\boldsymbol{\Lambda}_\mathrm{d}^\mathrm{neutral}$. All the existing k-Same methods as well as the k-Diff-furthest method operate with a set of face images and is not viable with a single face image. Furthermore, the original neutral face $\boldsymbol{\Lambda}^\mathrm{neutral}$ is not always available in real-life applications and searching for it in a video sequence is complicated and can be time consuming. To resolve these two issues, we calculate identity shift using the following equation instead:

$$\text{identity shift} = (\boldsymbol{\Lambda}_\mathrm{d}^* - \boldsymbol{\Lambda}^*), \tag{3}$$

where $\boldsymbol{\Lambda}^*$ is the nearest face to $\boldsymbol{\Lambda}^\mathrm{neutral}$ chosen from a person-specific set of neutral face images. This estimation has also been used in [7] with good experimental results. We name the set of face images $\{\boldsymbol{\Lambda}^*\}$ the donor set. In this work, the calculation of $\{\boldsymbol{\Lambda}_\mathrm{d}^*\}$ is carried out offline using the k-Diff-furthest method in a single pass. Although $\{\boldsymbol{\Lambda}_\mathrm{d}^*\}$ can be generated using any face de-identification method, k-Diff-furthest has been used here to maintain distinguishability between identities after de-identification. The identity shift $(\boldsymbol{\Lambda}_\mathrm{d}^* - \boldsymbol{\Lambda}^*)$ for each face $\boldsymbol{\Lambda}^*$ (or identity) in the donor set is also calculated offline to increase the efficiency of the proposed system. To minimise the initial delay in the online operation of the system, we always use the first available video frame of a person's face to choose its closest donor.

4.3 Feature normalisation and retainment of the original head pose

Our proposed approach uses a generic AAM face model to make sure it is capable of accurately representing any face instance in the input video. Our generic AAM model is trained using face images of various small head poses, facial expressions and illumination conditions. As a result, the first two shape dimensions of our trained AAM model represent the pitch and yaw of a face, while the first three texture dimensions represent the illumination of a face image. Considering that illumination and head pose have a noticeable impact on the accuracy of face recognition, we normalise the illumination, pitch and yaw of $\boldsymbol{\Lambda}^1$ (the first instance of $\boldsymbol{\Lambda}^i$) before searching for its nearest donor face $\boldsymbol{\Lambda}^*$. This normalisation is achieved by setting the parameters of the above-mentioned AAM dimensions to zero.

The same normalisation has also been applied to the set of donor faces such that the identity shift generated using (3) will not alter illumination, pitch and yaw of a face. Hence, through the calculation of (2), all these characteristics of an original face $\boldsymbol{\Lambda}^i$ will be automatically passed to its de-identified version of $\boldsymbol{\Lambda}_\mathrm{d}^i$. As for the translation and roll of $\boldsymbol{\Lambda}^i$, we restore them on $\boldsymbol{\Lambda}_\mathrm{d}^i$ through

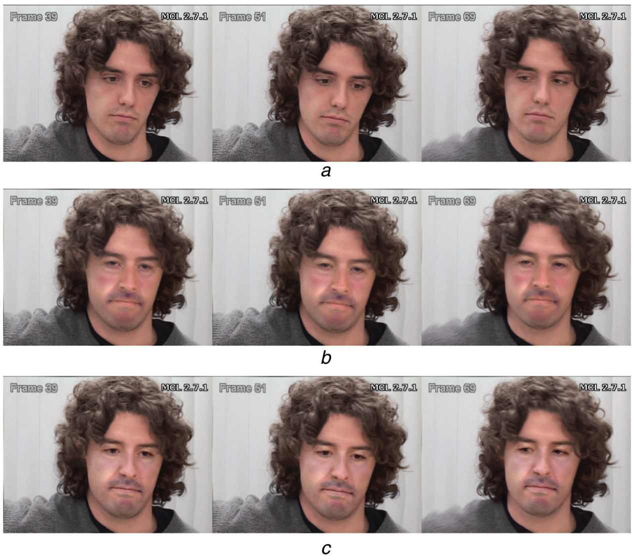

Fig. 4 *Result of both texture transfer approaches*
(a) Original video frame, ***(b)*** De-identified video frame whose texture has been transferred in the AAM feature space, ***(c)*** De-identified video frame whose texture has been transferred in raw pixel space. Figures (*b*) and (*c*) are using the same de-identified face shape which is different from (*a*). Figure (*c*) contains more texture details of (*a*), e.g. wrinkles and gaze

Procrustes analysis based on three facial landmarks – the inner corners of the eyes and the tip of the nose. The retainment of original illumination and head pose makes the whole video frame look much more natural when the de-identified face region is merged with the original image background. It also makes head movements in the de-identified sequence remain smooth.

4.4 Removal of glasses, beard and moustache through AAM face representation

In our proposed algorithm, the de-identified neutral faces $\boldsymbol{\Lambda}_{\mathrm{d}}^{*}$ in (3) are generated using the *k*-Diff-furthest method. As shown in Fig. 2, the *k*-Diff-furthest method de-identifies original faces in cluster $\boldsymbol{C}_{\mathrm{c}}$ by shifting them to around the centre of the opposite cluster $\boldsymbol{C}_{\mathrm{f}}$ and vice versa. The existence of glasses or heavy facial hair on even one original face in a cluster will lead to a faded version of the same artefact showing on its cluster centre and subsequently all the de-identified faces around this centre. To ensure good visual quality with our de-identified faces, we remove these artefacts from the face images in the donor set before applying the *k*-Diff-furthest method. In this work, the removal of these artefacts is achieved through excluding face images with glasses or noticeable facial hair from the training set of our generic AAM model. A statistical face model such as an AAM can only represent features that it has seen in its training set. As a result, glasses or heavy facial hair on any unseen image will be automatically removed when the image is projected into our specially trained AAM feature space.

4.5 Implementing face de-identification in different feature space

In AAM, the shape and texture of a face can be represented and processed separately. To ensure perfect alignment, we perform de-identification of face shapes in the AAM model space. For the de-identification of face texture, two different approaches have been attempted. The first approach completes the entire calculation of (2) in the AAM feature space and then reconstructs the image pixels based on the de-identified AAM features. The second approach converts the identity shift calculated in the AAM space into image pixels offline in advance and adds the resulting image to the frames of the input video in real time. These two approaches have similar real-time computational load with the first approach involving one more matrix multiplication. Fig. 4 shows the result of both approaches with three example video frames. As shown, the de-identified video frame generated by the second approach looks more natural than the first approach. However, the second approach does not project the original faces into the AAM space, so any artefacts such as glasses or a mole on the face will be passed from the original image to its de-identified image and increase the re-identification risk. The re-identification risk of the de-identified images produced by these two different approaches is evaluated and compared in Section 5.3.

4.6 Merging with the original background

The main challenge of merging a de-identified face region with its original background is given by the noticeable differences between the two in terms of skin tone, illumination, direction of lighting etc. In our work, the method of Poisson seamless cloning [23] has been used to achieve good visual quality of the blended images at the cost of relatively long processing time.

As the de-identification process alters the shape of the face, the background of the original image has to be deformed to fit the de-identified face region. In this work, the deformation of the background is achieved using moving least squares [24].

5 Experiments

Our proposed face de-identification system has been tested on the video sequences from the UNBC-McMaster Shoulder Pain Expression Archive Database [25]. The UNBC-McMaster database contains 200 video sequences of the faces of 25 subjects, where faces in a video all relate to the same subject. We have selected 184 video sequences from the database to make sure a complete face with the full set of 68 facial landmarks is detected in each frame. There are on average 238 frames per video.

In our work, the donor set is composed of 780 near neutral faces from the FERET dataset [26] that are also near frontal. Before applying face de-identification to video sequences, de-identification of the donor set is carried out offline in a single pass and so is the calculation of the identity shift for each donor face. The *k*-Diff-furthest method has been used to de-identify the donor set unless specified otherwise as in Section 5.3.

To enable an accurate representation of the faces in the donor set as well as those in the test video sequences, we train a generic AAM with 1952 near frontal faces from the FERET dataset. As stated in Sections 4.3 and 4.4, these faces present various facial expressions, illumination and small head poses and are without glasses or heavy facial hair. After principal component analysis of the training set, eight-shape components are kept to represent 90% of the shape variance within the training set and 59 texture components for 90% of the texture variance.

In this work, various experiments have been conducted to evaluate the proposed approach's ability to address the four challenges of face de-identification in videos. The rest of this section discusses the results of these experiments.

5.1 Preservation of facial expressions

As facial expressions are very complicated and multiple expressions often appear on a face simultaneously, e.g. happily surprised or sadly shocked. In the study of facial expression analysis, action units (AUs) on human faces have been widely used to describe facial expressions [27] due to their effectiveness. The UNBC-McMaster dataset contains the ground truth of nine AUs including brow-lowering (AU4), cheek-raising (AU6), eyelid tightening (AU7), nose wrinkling (AU9), upper-lip raising (AU10), oblique lip raising (AU12), horizontal lip stretch (AU20), lips parting (AU25) and jaw-dropping (AU26), where the intensity of each AU has been scored manually as an integer from 0 to 6 inclusively. To test the expression preservation performance of our system, the intensity level of each AU is compared between the original video frames and their corresponding de-identified frames. The AU intensity of a video frame is predicted by OpenFace [21], which generates AU intensity as real numbers.

Fig. 5 shows the average absolute difference between the AU intensities calculated by OpenFace and the ground truth for the original video frames as well as the de-identified frames generated

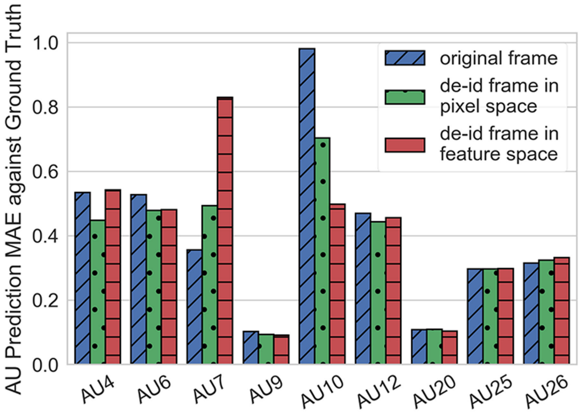

Fig. 5 *Mean absolute errors of OpenFace AU detection results of the original frames and the de-identified frames, in comparison with the AU intensity ground truth provided by the dataset*

in the pixel space and the AAM feature space, respectively. The difference values are averaged over all the 43,734 test video frames. Fig. 5 shows that, apart from AU7 eyelid tightener and AU10 upper lip raiser, the AU intensity values of the de-identified frames have remained almost the same as of their original frames. For AU10, the AU intensity values predicted from the de-identified video frames are closer to the ground truth than their original video frames. Even in the worst case of AU7, the AU intensity has remained within the same integer level after face de-identification.

As shown by the blue bars in Fig. 5, OpenFace is not always accurate and have found some AUs hard to predict. Considering this, we have performed the same AU intensity comparison test with only the frames that OpenFace can predict correctly, i.e. when the rounded-up value of OpenFace's prediction matches with the ground truth values. Fig. 6 shows the comparison results. Again, there is hardly any change in AU intensity for AU9 nose wrinkler and AU20 lip stretcher. The highest difference is still with AU7 but at a negligible level of 0.3 out of 6. Example video frames in Fig. 6 are included to demonstrate the visual impact of the average intensity difference obtained for each AU. As seen from these example frames, there is hardly any visual difference in the facial expressions before and after face de-identification.

5.2 Identity consistency of the de-identified videos

Each UNBC-McMaster video sequence contains face instances of only one person. To evaluate identity consistency within each de-identified video, i.e. to test whether the face instances within a de-identified video still all relate to the same identity, we have de-identified every UNBC-McMaster video independently and conducted a ten-fold cross validation on the de-identified faces using the $\mathrm{LBP}^{u2}_{8,2}$ face descriptor in [28]. The face images used in this experiment have been cropped out, aligned with the inner corners of the eyes and the tip of the nose, and resized to 100 × 100 (see Fig. 7 for some examples). The result shows that 99.98% of our de-identified faces are top-rank matched as another de-identified face from the same video and therefore of the same original identity.

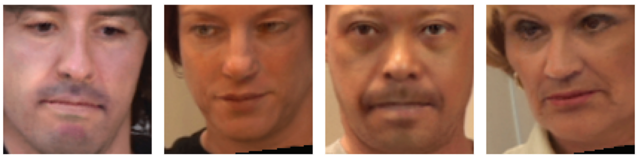

Fig. 7 *Examples of the cropped and resized face images used in our privacy protection test*

We have also evaluated our method's capability of preserving identity consistency across videos. Here, we grouped UNBC-McMaster videos according to their identities given by the dataset and used the same identity shift to de-identify all the videos in the same group. The ten-fold cross validation result shows that 99.99% of the de-identified faces are top-rank matched as another de-identified face from the same identity group, where 97.57% of the correct matches are found within the same video and 2.43% are found from another video of the same original identity.

5.3 Privacy protection ability

De-identification of the donor set can be done using any face de-identification method and we have tested a few including the *k*-Diff-furthest method, the benchmark *k*-Same method and three face swapping methods. The three face swapping methods follow the study in [29], where Rank-*i* face swapping replaces an original face with its *i*th closest face chosen from a donor set of size *N*. Our donor set consists of *N* = 780 faces from the FERET dataset. All 43,734 original faces extracted from the UNBC-McMaster videos plus our donor set have been used to form the face gallery in the re-identification test. The donor set has been included in the gallery to increase the number of subjects. Otherwise, the re-identification risk even by random matching would be 1 out of the 25 subjects of the UNBC-McMaster database. Eigenfaces [30] and LBP [28] face recognition software have been used to match the de-identified faces of the UNBC-McMaster videos with those in the face gallery. For each face de-identification method being tested, all the 43,734 de-identified faces before Poisson blending and background merging have been used as the probe images. Again, the face regions are cropped out, aligned and resized to 100 × 100.

Difference in AU intensity	AU4	AU6	AU7	AU9	AU10	AU12	AU20	AU25	AU26
	0.148	0.232	0.302	0.06	0.26	0.215	0.077	0.135	0.166

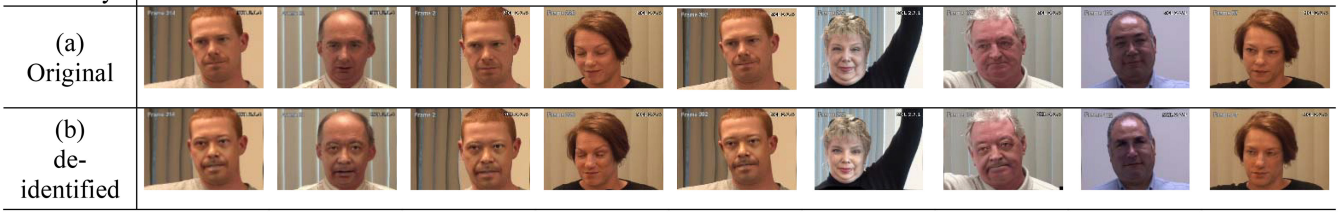

Fig. 6 *Average absolute difference in AU intensity between original frames and their de-identified frames, with example frames demonstrating that the average difference with each AU causes hardly any visible changes to the facial expressions*

Table 1 Rank-1 re-identification risk of the de-identified video frames

	Texture transfer in feature space					Texture transfer in pixel space				
	k-same, %	*k*-DF[a], %	Rank-1, %	Rank-200, %	Rank-*N*, %	*k*-same, %	*k*-DF[a], %	Rank-1, %	Rank-200, %	Rank-*N*, %
eigenfaces	3.83	2.56	4.78	2.67	0.51	23.28	16.76	21.20	19.29	4.66
LBP	2.66	1.29	1.98	1.74	0.95	20.95	13.59	19.36	13.09	8.33

[a] *k*-DF stands for *k*-Diff-furthest.

Table 1 shows the rank-1 re-identification risk of the de-identified faces and compares the privacy protection performance of the above-mentioned face de-identification methods as well as the two texture transfer approaches described in Section 4.5. Rank-1 re-identification risk in Table 1 relates to the cases when a de-identified video frame has been matched with any frame from its original video sequence.

Results in Table 1 show that when implementing (2) in the AAM feature space all the tested face de-identification methods have been able to provide sufficient privacy protection against similarity-based face recognition software. Transferring the face texture in pixel space gives the de-identified faces more visual details of the original facial expression. However, as expected, it sacrifices the privacy protection performance as more original face texture details such as wrinkles are also transferred to the de-identified faces.

As shown in Table 1, the *k*-Diff-furthest method has outperformed the *k*-Same method in both the AAM feature space and the pixel space. The performance of Rank-1 face swapping is comparable to that of *k*-Same. With our donor set of 780 faces, the performance of *k*-Diff-furthest is comparable to Rank-200 face swapping. Rank-*N* face swapping has generated the lowest re-identification risk. However, Rank-*N* face swapping tends to choose the same outliers in the donor set to replace the original faces. In our experiments, Rank-*N* has used only 14 faces to replace the entire donor set of 780 faces with 625 faces (80%) sharing the same five new identities. The mechanism of the *k*-Diff-furthest method guarantees that each original face is replaced with a unique de-identified face and the set of de-identified faces remains as diverse and distinguishable as the original set [6]. Furthermore, both the *k*-Same and *k*-Diff-furtheset methods replace an original face with a synthesised face while the three face swapping methods use a natural face chosen from a donor set.

5.4 System efficiency

We have run our experiments in MATLAB 2016a on an Intel Core i7-4770 CPU at 3.40 GHz. It takes on average 0.082 s per video frame (i.e. 12.2 frames/s) to extract an original face instance, generate the de-identified face and blend it into the background of the original frame. The face de-identification step defined in (2), i.e. the simple addition, takes <1% of the overall processing time, face warping to combine the de-identified face shape and texture takes 25%, Poisson blending 10% and background warping 25%. According to the comparison study presented in [31], MATLAB is between nine and 11 times slower than the best C++ executable. According to [32], Pyrex/Cython is 12 times faster than MATLAB for solving Laplace's equation and C++ is 13.4 times faster. It is easy to see that with a more efficient programming language such as C++, the proposed system can easily support real-time face de-identification in videos at the highest standard video frame rate of 60 frames/s.

5.5 Further remarks

In this section, we have shown that the proposed approach is able to achieve expression preservation with the particular set of test videos. However, it is worth mentioning that these results depend on accurate annotation of the facial landmarks and this can be challenging for faces with large poses. Furthermore, all video frames considered in our experiments display a full set of landmarks. When the head pose becomes so large that some of the facial landmarks become invisible, additional steps are required to establish landmark correspondence between the identity shift and the original video frame in order to implement the addition of the two as defined in (2).

In our proposed method, identity consistency is achieved by applying the same identity shift to all face instances of the same person. However, in practice, the precise identities of the original face instances are not always known and the scene may switch between characters (identities) throughout a sequence. When this is the case, the identity consistency performance of our approach will heavily depend on the accuracy of the face identification method employed.

The basis of privacy protection of our proposed approach is the *k*-Diff-furthest method, which demands no overlap between the pair of clusters formed in each iteration. This condition cannot be guaranteed by the subsequent FET process in our approach. Although our experimental results show good privacy protection performance, our proposed approach does not provide theoretical guarantee of anonymity. Finally, while the proposed approach seeks to thwart face recognition software, correct identification is still possible by recognising, for example clothes, behaviour, or contextual information.

6 Conclusion

To address the challenges of face de-identification in videos, we have proposed a highly efficient approach that achieves privacy protection and preservation of facial expressions simultaneously through the simple operation of adding a pre-calculated identity shift to the original face instances in the input video. The use of the same identity shift for each subject in the original videos guarantees identity consistency in the de-identified video sequences. It also allows the dynamics of facial expressions presented in an original video to be preserved in the de-identified video. Computation time analysis has shown that the proposed approach can be used to perform face de-identification on videos at the highest standard frame rate.

7 References

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

[2] Newton, E.M., Sweeney, L., Malin, B.: ‘Preserving privacy by de-identifying face images’, *IEEE Trans. Knowl. Data Eng.*, 2005, **17**, (2), pp. 232–243

[3] Gross, R., Sweeney, L., de la Torre, F., *et al.*: ‘Model-based face de-identification’. Proc. Conf. Computer Vision and Pattern Recognition Workshop, New York, USA, 2006, pp. 161–161

[4] Gross, R., Airoldi, E., Malin, B., *et al.*: ‘Integrating utility into face de-identification’. Proc. 5th Int. Conf. Privacy Enhancing Technologies, Berlin, Heidelberg, 2005, pp. 227–242

[5] Meng, L., Sun, Z.: ‘Face de-identification with perfect privacy protection’. Proc. 37th Int. Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), May 2014, pp. 1234–1239

[6] Sun, Z., Meng, L., Ariyaeeinia, A.: ‘Distinguishable de-identified faces’. Proc. 11th IEEE Int. Conf. Workshop on Automatic Face and Gesture Recognition, Ljubljana, Slovenia, May 2015, pp. 1–6

[7] Meng, L., Sun, Z., Ariyaeeinia, A., *et al.*: ‘Retaining expressions on de-identified faces’. Proc. 37th Int. Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), May 2014, pp. 1252–1257

[8] Cootes, T.F., Edwards, G.J., Taylor, C.J.: ‘Active appearance models’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2001, **23**, (6), pp. 681–685

[9] Lucey, S., Matthews, I., Hu, C., *et al.*: ‘AAM derived face representations for robust facial action recognition’. Proc. 7th Int. Conf. Automatic Face and Gesture Recognition, Southampton, UK, 2006, pp. 155–162

[10] de la Hunty, M., Asthana, A., Goecke, R.: ‘Linear facial expression transfer with active appearance models’. Proc. 20th Int. Conf. Pattern Recognition, Istanbul, Turkey, August 2010, pp. 3789–3792

[11] Theobald, B.-J., Matthews, I., Mangini, M., *et al.*: ‘Mapping and manipulating facial expression’, *Lang. Speech*, 2009, **52**, (Pt 2–3), pp. 369–386

[12] Thies, J., Zollhöfer, M., Nießner, M., *et al.*: ‘Real-time expression transfer for facial reenactment’, *ACM Trans. Graph.*, 2015, **34**, (6), pp. 1–14

[13] Thies, J., Zollhofer, M., Stamminger, M., *et al.*: ‘Face2face: real-time face capture and reenactment of RGB videos’. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Las Vegas, USA, June 2016, pp. 2387–2395

[14] Cristinacce, D., Cootes, T.: ‘Automatic feature localisation with constrained local models’, *Pattern Recognit.*, 2008, **41**, (10), pp. 3054–3067

[15] Xiong, X., De la Torre, F.: ‘Supervised descent method and its applications to face alignment’. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Portland, USA, June 2013, pp. 532–539

[16] Kazemi, V., Sullivan, J.: ‘One millisecond face alignment with an ensemble of regression trees’. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Columbus, USA, June 2014, pp. 1867–1874

[17] Ren, S., Cao, X., Wei, Y., *et al.*: ‘Face alignment at 3000 FPS via regressing local binary features’. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Columbus, USA, June 2014, pp. 1685–1692

[18] Baltrusaitis, T., Robinson, P., Morency, L.-P.: ‘Constrained local neural fields for robust facial landmark detection in the wild’. Proc. IEEE Int. Conf. Computer Vision Workshop, Sydney, Australia, December 2013, pp. 354–361

[19] Gross, R., Matthews, I., Cohn, J., *et al.*: ‘Multi-PIE.’, *Image Vis. Comput.*, 2010, **28**, (5), pp. 807–813

[20] Sagonas, C., Antonakos, E., Tzimiropoulos, G., *et al.*: ‘300 faces in-the-wild challenge: database and results’, *Image Vis. Comput.*, 2016, **47**, pp. 3–18

[21] Baltrusaitis, T., Robinson, P., Morency, L.-P.: ‘Openface: an open source facial behavior analysis toolkit’. Proc. IEEE Winter Conf. Applications of Computer Vision, Lake Placid, USA, March 2016, pp. 1–10

[22] Sun, Z., Meng, L., Ariyaeeinia, A., *et al.*: 'Privacy protection performance of de-identified face images with and without background'. Proc. 39th Int. Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, May 2016, pp. 1354–1359

[23] Pérez, P., Gangnet, M., Blake, A.: 'Poisson image editing', *ACM Trans. Graph.*, 2003, **22**, (3), pp. 313–318

[24] Schaefer, S., McPhail, T., Warren, J.: 'Image deformation using moving least squares', *ACM Trans. Graph.*, 2006, **25**, (3), pp. 533–540

[25] Lucey, P., Cohn, J.F., Prkachin, K.M., *et al.*: 'Painful data: the UNBC-McMaster shoulder pain expression archive database'. Proc. IEEE Int. Conf. Automatic Face & Gesture Recognition Workshop, Santa Barbara, USA, March 2011, pp. 57–64

[26] Phillips, P.J., Rizvi, S.A., Rauss, P.J.: 'The FERET evaluation methodology for face-recognition algorithms', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2000, **22**, (10), pp. 1090–1104

[27] Cohn, J., Ambadar, Z., Ekman, P.: 'Observer-based measurement of facial expression with the facial action coding system', Coan, J.A., Allen, J.J.B. (Eds.): '*The handbook of emotion elicitation and assessment*' (Oxford University Press, 2006), pp. 203–221

[28] Ahonen, T., Hadid, A., Pietikainen, M.: 'Face description with local binary patterns: application to face recognition', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006, **28**, (12), pp. 2037–2041

[29] Bitouk, D., Kumar, N., Dhillon, S., *et al.*: 'Face swapping: automatically replacing faces in photographs', *ACM Trans. Graph.*, 2008, **27**, (3), pp. 39:1–39:8

[30] Turk, M., Pentland, A.: 'Eigenfaces for recognition', *J. Cogn. Neurosci.*, 1991, **3**, pp. 71–86

[31] Aruoba, S.B., Fernández-Villaverde, J.: 'A comparison of programming languages in economics'. NBER Working Papers 20263, National Bureau of Economic Research, Cambridge, MA, USA, 2014

[32] 'A beginner's guide to using Python for performance computing', https://scipy.github.io/old-wiki/pages/PerformancePython, accessed December 2016