A Spatial Self-Similarity Based Feature Learning Method for Face Recognition under Varying Poses

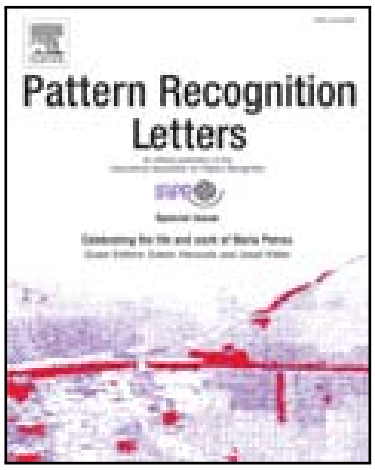
Duan, Xiaodong; Tan, Zheng-Hua

Published in:
Pattern Recognition Letters

DOI (link to publication from Publisher):
10.1016/j.patrec.2018.05.007

Publication date:
2018

Document Version
Accepted author manuscript, peer reviewed version



A Spatial Self-Similarity Based Feature Learning Method for Face Recognition under Varying Poses

Xiaodong Duan, Zheng-Hua Tan

PII: S0167-8655(18)30173-9
DOI: 10.1016/j.patrec.2018.05.007
Reference: PATREC 7173

To appear in: *Pattern Recognition Letters*

Received date: 2 June 2017
Revised date: 8 March 2018
Accepted date: 4 May 2018

Please cite this article as: Xiaodong Duan, Zheng-Hua Tan, A Spatial Self-Similarity Based Feature Learning Method for Face Recognition under Varying Poses, *Pattern Recognition Letters* (2018), doi: 10.1016/j.patrec.2018.05.007

Highlights

- Propose a simple but e ective method for pose varying face recognition.
- The method has no need for pose information.
- Pose related part in a local feature is removed by a linear transformation.
- The linear transformation is learned through a closed-form solution.

Pattern Recognition Letters
journal homepage: www.elsevier.com

A Spatial Self-Similarity Based Feature Learning Method for Face Recognition under Varying Poses

Xiaodong Duan**, Zheng-Hua Tan

Department of Electronic Systems, Aalborg University, Fredrik Bajers Vej 7B, Aalborg, DK-9220

ABSTRACT

In this paper, we propose a low-complexity method to learn pose-invariant features for face recognition with no need for pose information. In contrast to the commonly used approaches of recovering frontal face images from profile views, the proposed method extracts the subject related part from a local feature by removing its pose related part. First, the method generates a self-similarity feature by computing the distances between local feature descriptors of different non-overlapping blocks in a face image. Secondly, it subtracts from the local feature a linear transformation of the self-similarity feature and the transformation matrix is learned through minimizing the feature distance between face images from the same person but under different poses while retaining the discriminative information across different persons. In order to evaluate our method, extensive experiments on face recognition across poses are conducted using FERET and Multi-PIE, in addition, experiments on face recognition under unconstrained situations are conducted using LFW-a. Results on these three public databases show that the proposed method is able to significantly improve the recognition performance as compared with using the original local features and outperforms or is comparable to related, state-of-the-art pose-invariant face recognition approaches.

1. Introduction

Face recognition has a broad range of applications such as biometric authentication, surveillance and human robot interaction, to name a few. As a non-intrusive authentication technique, face recognition is commonly deployed for security especially in situations where people do not cooperate. By recognizing the face of a person, a robot can provide personalized interactions and services. Therefore, face recognition has attracted significant attention in computer vision and machine learning communities during past few decades. However, recognizing faces under varying poses remains a challenging task [1, 2, 3, 4].

In this paper, we focus on improving the robustness of face recognition against pose variations. A great number of face recognition methods have been proposed to tackle the problem caused by pose variations. These methods can be categorized into three groups: pose-invariant 2D and 3D methods [5] and deep learning based methods [1].

In order to achieve good performance, pose-invariant 2D approaches often require pose information, which is obtained either manually or by a pose estimation method. Chai et al. propose a method in which a virtual frontal view is generated from a nonfrontal view through locally linear regression [6]. Similarly, markov random fields are used to generate virtual frontal face images in [7]. Sharma et al. use partial least squares to map 2D images under various poses to a linear subspace for face recognition with different poses [8, 9]. In [10], a face image is represented by a linear combination of training images under the same pose, and then the obtained coefficients are used for face recognition.

3D approaches are applied on face recognition to avoid information loss [11] when converting a 3D world to a 2D image. In [12], pose variations are handled by face symmetry with high quality 3D data captured by a 3D scanner. Although these 3D methods are able to obtain favorable performance, their computational complexity is usually high, e.g., the average speed for processing a face scan is approximately 18 seconds in [12]. Further, it is infeasible to obtain 3D data in some application scenarios.

Due to the impressive performance of deep learning [13] in various applications, a number of deep learning based methods are proposed for pose-invariant face recognition. The deep

**This work is supported by the Danish Council for Independent Research | Technology and Production Sciences under grant number: 1335-00162 (iSocioBot).

**Corresponding author: Tel.: +4599403847; fax: +4598151583;
e-mail: xd@es.aau.dk (Xiaodong Duan)

learning structure in [14] consists of two modules, one to learn a Face Identity-Preserving (FIP) feature while the other to reconstruct the frontal face image from FIP. It takes profile images as input and frontal images as output. In [15], stacked progressive autoencoders are used to learn a pose-robust feature. Progressively, each autoencoder layer tries to construct face images with a smaller pose angle than the previous layer with frontal face images in the output layer. The last hidden layer is used as the face recognition feature. Convolutional Neural Networks (CNNs) are applied for face recognition in [16] and [17] using 2.6 millions and 0.7 million training images, respectively. In general, deep learning based methods have a large number of parameters to learn, indicating the need for a large amount of training data and a high computation cost.

In this paper, a low-complexity feature learning method is proposed for pose-invariant face recognition with no need for any prior knowledge about pose angles, which is an extension of [18]. This is motivated by the following observation. In most cases, humans can easily recognize a person through a face image of the person under different poses or even from a sketch of her/his face. Therefore, one can assume that there is certain subject-related information in a face image invariant to pose variations and being enough for face recognition. We can further assume that subject-related information is mostly encoded in features for face recognition. Quite likely, unfortunately, these features simultaneously capture pose variations of face images, which can make the distance between features from two different persons but with the same pose smaller than the distance between features from the same person but with different poses, leading to a misclassification. In order to overcome this problem, we propose a method aiming at removing the pose related part from a feature while retaining the subject related part. Since the subject related part is obtained by subtracting the pose related part from a feature, we refer the new feature as a subtracted feature.

In our method, a local feature is first extracted from a face image and then a spatial self-similarity feature is computed from the local feature, which is different from [18]. This makes it more robust to noises in a face image as compared with the similarity feature [18] generated from the gray image directly. The purpose of computing the self-similarity feature is to capture the spatial symmetrical information of a face image so as to embed pose variations in it. The self-similarity feature is further transformed to be a pose related part of the local feature through a linear transformation. This linear transformation is learned through minimizing the distance between the subtracted features of the same person with different poses while moving the mean subtracted feature of each person apart from each other, not through making the subtracted features closer to the mean feature vector of face images of each person under different poses as done in [18]. In the end, a subtracted feature is obtained by subtracting the linearly transformed self-similarity feature from the local feature. Extensive experiments on three publicly available databases are conducted and results show the promising performance of the proposed method. Since only a linear transformation needs to be learned during training and applied during evaluation, the method is computationally efficient, as opposed to existing methods of converting profile face images to frontal ones.

2. Proposed method

Local features are widely used for face recognition. To extract a local feature a face image is usually first segmented into K non-overlapping blocks. Then, on each block, a feature descriptor vector $\boldsymbol{l}^i, i = 1, \cdots, K$, e.g., local phase quantization (LPQ) and scale-invariant feature transform (SIFT), is computed. These descriptors $\boldsymbol{l}^i$s are concatenated to form a feature $\boldsymbol{l}$ of the face image. However, a feature $\boldsymbol{l}$ naturally encodes not only useful subject related information but also unwanted pose variations. In other words, a feature consists of two parts, a subject related part and a pose related part. For face recognition, it is desirable to remove the pose related part while retaining the subject related part. In order to derive a low-complexity and effective solution and motivated by the success of metric learning methods, we assume that a local feature is a linear combination of the subject related part $\boldsymbol{s}$ and the pose related part $\boldsymbol{p}$ as follows:

$$\boldsymbol{l} = \boldsymbol{s} + \boldsymbol{p}. \tag{1}$$

If we know $\boldsymbol{p}$, $\boldsymbol{s}$ is then obtained by simply subtracting $\boldsymbol{p}$ from $\boldsymbol{l}$.

Pose variations can greatly change the symmetrical information of a face image while a frontal face image is generally spatially symmetrical. Therefore, a feature representing the symmetrical information could be used to represent the pose related information. In other words, it is possible to extract the pose related information from a symmetry feature. In our method, this symmetric information is captured by the spatial self-similarity feature $\boldsymbol{v}$. The feature $\boldsymbol{v}$ is generated from $\boldsymbol{l}$, which is described in Section 2.1.

In recent years, metric learning methods [19, 20] have attracted much attention and have shown encouraging performance on a variety of applications. These methods are mostly a linear transformation of feature vectors. In deriving the pose related part, we use the similar concept and thus assume that the pose related part can be obtained by linearly transforming $\boldsymbol{v}$ as follows:

$$\boldsymbol{p} = \boldsymbol{A}\boldsymbol{v}, \tag{2}$$

where only the transformation matrix $\boldsymbol{A}$ needs to be learned from training data, leading to a low-complexity solution. The learning method is presented in Section 2.2.

2.1. Spatial self-similarity feature

Since a local feature $\boldsymbol{l}$ is concatenated by the feature descriptor vectors $\boldsymbol{l}^i$s , which are extracted on non-overlapping blocks, we compute spatial self-similarity feature from the local feature. An element v^j of a self-similarity feature vector $\boldsymbol{v}$ is computed as

$$v^j = -f(\boldsymbol{l}^i, \boldsymbol{l}^k), i = 1, \cdots, K-1, k = i+1, \cdots, K, \tag{3}$$

where j is used to index the scalar element of $\boldsymbol{v}$, K is the number of the non-overlapping blocks and $j = (i-1)K + (k-i)$. f is a function to compute the distance between two vectors, e.g.,

1-norm or 2-norm distance. v^j is a scalar and the dimension of $\boldsymbol{v}$ is calculated as below:

$$|\boldsymbol{v}| = K \times (K-1)/2. \tag{4}$$

This self-similarity vector is capable of capturing the spatial similarities between different blocks, which makes it as a proper representation of the spatial symmetrical information of a face image. Furthermore, since this vector is generated from a feature rather than from raw image pixel intensity as done in [18], it is more robust to noises in face images.

2.2. Learning algorithm

Substitute equation 2 into equation 1, the subject related part vector can be obtained as following:

$$\boldsymbol{s} = \boldsymbol{l} - \boldsymbol{A}\boldsymbol{v}, \tag{5}$$

where only the transformation matrix $\boldsymbol{A}$ needs to be learned from training data. Since $\boldsymbol{s}$ is obtained by a subtraction operation, we refer it as a subtracted feature vector. The learning goals of this linear transformation are 1) to remove the pose related part in a local feature vector $\boldsymbol{l}$ and 2) to maintain the discriminative capability of a subtracted feature vector for face recognition.

For the training data, let us denote M the number of different persons and N the number of different poses for each person, which gives $M \times N$ training images in total. An image is denoted by x.

To meet the first goal, a training pair dataset is constructed as

$$S = \left\{(x_{i,j}, x_{i,k})\right\}, \tag{6}$$

where $i = 1, \cdots, M$, $j = 1, \cdots, N-1$ and $k = j+1, \cdots, N$. This means that S consists of image pairs, each of which is a pair of images from the same person under two different poses. The distance between two subtracted feature vectors, $\boldsymbol{s}_{i,j}$ of $x_{i,j}$ and $\boldsymbol{s}_{i,k}$ of $x_{i,k}$, should be as small as possible since $(x_{i,j}, x_{i,k}) \in S$ and there is no pose related part in the subtracted vectors. Therefore, we define the cost function to be minimized as follows:

$$J_1 = \sum_{(x_{i,j}, x_{i,k}) \in S} \left\| \boldsymbol{s}_{i,j} - \boldsymbol{s}_{i,k} \right\|_2^2. \tag{7}$$

In order to meet the second goal, we construct another training pair dataset consisting of pairs of image from two different persons with all poses as follows:

$$D = \{(x_{m,:}, x_{n,:})\}, \tag{8}$$

where $m = 1, \cdots, M-1$ and $n = m+1, \cdots, M$. The mean of subtracted feature vector $\hat{\boldsymbol{s}}$ for a person m is then computed as

$$\hat{\boldsymbol{s}}_m = \frac{\sum_{k=1}^{N} (\boldsymbol{s}_{m,k})}{N}. \tag{9}$$

Similarly, the mean feature vector $\hat{\boldsymbol{l}}_m$ and mean self-similarity vector $\hat{\boldsymbol{v}}_m$ for person m are

$$\hat{\boldsymbol{l}}_m = \frac{\sum_{k=1}^{N} (\boldsymbol{l}_{m,k})}{N}, \tag{10}$$

and

$$\hat{\boldsymbol{v}}_m = \frac{\sum_{k=1}^{N} (\boldsymbol{v}_{m,k})}{N}. \tag{11}$$

Using training pair dataset D, we define a second cost function as

$$J_2 = \sum_{(x_{m,:}, x_{n,:}) \in D} \|\hat{\boldsymbol{s}}_m - \hat{\boldsymbol{s}}_n\|_2^2. \tag{12}$$

Since the distance between the subtracted feature vectors of different persons should be as large as possible in order to maintain its discriminative capability, equation 12 should be maximized. This cost function is the summation of the mean of subtracted feature vectors of each person rather the than subtracted feature vector of each different pose for three reasons. First, this avoids the need for pose information when constructing D, which makes our method pose-free. Secondly, averaging data across poses is also regarded as noise reduction processing. Thirdly, it reduces the number of pairs in D.

In the end, the cost function used to estimate the transformation matrix $\boldsymbol{A}$ is formulated through combining J_1 and J_2 as follows:

$$\begin{aligned} J &= J_1 - \alpha \times J_2 \\ &= \sum_{(x_{i,j}, x_{i,k}) \in S} \left\| \boldsymbol{s}_{i,j} - \boldsymbol{s}_{i,k} \right\|_2^2 - \alpha \times \sum_{(x_{m,:}, x_{n,:}) \in D} \|\hat{\boldsymbol{s}}_m - \hat{\boldsymbol{s}}_n\|_2^2 \\ &= \sum_{(x_{i,j}, x_{i,k}) \in S} \left\| (\boldsymbol{l}_{i,j} - \boldsymbol{l}_{i,k}) - \boldsymbol{A}(\boldsymbol{v}_{i,j} - \boldsymbol{v}_{i,k}) \right\|_2^2 \\ &\quad - \alpha \times \sum_{(x_{m,:}, x_{n,:}) \in D} \left\| (\hat{\boldsymbol{l}}_m - \hat{\boldsymbol{l}}_n) - \boldsymbol{A}(\hat{\boldsymbol{v}}_m - \hat{\boldsymbol{v}}_n) \right\|_2^2, \end{aligned} \tag{13}$$

where α is a scalar weighting factor. The purpose of J_2 in the equation 13 is to prevent the potential over-fitting problem of J_1 and maintain the discriminative capability of the subtracted feature. Some subject related information could be removed if over-fitting happened in J_1, which could decrease the performance. However, if too much weight is put on J_2, it is possible to move the subtracted feature vector to an arbitrary place in the feature space, which could be far away from the original feature and thus decrease the performance instead of improving it. Therefore, α only takes a small positive value.

The transformation matrix $\boldsymbol{A}$ can be estimated through minimizing equation 13. In order to do this, four matrices are formed as following:

$$\begin{aligned} \boldsymbol{L}_{:,o} &= \boldsymbol{l}_{i,j} - \boldsymbol{l}_{i,k} \mid (x_{i,j}, x_{i,k}) \in S \\ \boldsymbol{V}_{:,o} &= \boldsymbol{v}_{i,j} - \boldsymbol{v}_{i,k} \mid (x_{i,j}, x_{i,k}) \in S \\ \hat{\boldsymbol{L}}_{:,o} &= \hat{\boldsymbol{l}}_m - \hat{\boldsymbol{l}}_n \mid (x_{m,:}, x_{n,:}) \in D \\ \hat{\boldsymbol{V}}_{:,o} &= \hat{\boldsymbol{v}}_m - \hat{\boldsymbol{v}}_n \mid (x_{m,:}, x_{n,:}) \in D, \end{aligned} \tag{14}$$

based on which, equation 13 can be rewritten as

$$\begin{aligned} J = &\operatorname{Tr}\left((\boldsymbol{L} - \boldsymbol{A}\boldsymbol{V})^T(\boldsymbol{L} - \boldsymbol{A}\boldsymbol{V})\right) \\ &- \alpha \times \operatorname{Tr}\left((\hat{\boldsymbol{L}} - \boldsymbol{A}\hat{\boldsymbol{V}})^T(\hat{\boldsymbol{L}} - \boldsymbol{A}\hat{\boldsymbol{V}})\right). \end{aligned} \tag{15}$$

Taking the derivative of equation 15 with regard to $\boldsymbol{A}$, we can obtain $\nabla_{\boldsymbol{A}} J$ as

$$\nabla_{\boldsymbol{A}} J = -(\boldsymbol{L} - \boldsymbol{A}\boldsymbol{V})\boldsymbol{V}^T + \alpha \times (\hat{\boldsymbol{L}} - \boldsymbol{A}\hat{\boldsymbol{V}})\hat{\boldsymbol{V}}^T. \tag{16}$$

For a large amount of training data, to avoid high-complexity computations a gradient descent method could be applied to estimate $\boldsymbol{A}$ using equation 16. For a moderate size of training dataset, e.g., those used in this paper, we can set equation 16 be equal to zero, and then a closed-form solution is obtained for $\boldsymbol{A}$ as follows:

$$\boldsymbol{A} = \left(\boldsymbol{V}\boldsymbol{V}^T - \alpha \times \hat{\boldsymbol{V}}\hat{\boldsymbol{V}}^T\right)^{-1}\left(\boldsymbol{L}\boldsymbol{V}^T - \alpha \times \hat{\boldsymbol{L}}\hat{\boldsymbol{V}}^T\right), \qquad (17)$$

which is used for the experiments in Section 3.

2.3. Face recognition

For face recognition, a simple K-Nearest Neighbor (K-NN) method is applied. The Euclidean distance between the normalized subtracted vectors,

$$d_{mn} = \left\| \frac{\mathbf{s}_m}{\|\mathbf{s}_m\|_2} - \frac{\mathbf{s}_n}{\|\mathbf{s}_n\|_2} \right\|_2^2, \qquad (18)$$

is used as the distance metric of K-NN, where $\mathbf{s}_m$ is the subtracted vector of face image m in the gallery set and $\mathbf{s}_n$ is the subtracted vector of face image n in the probe set. K nearest neighbors are specified based on the distance calculated by equation 18 for each probe face image. The probe image takes the label that has the majority in these K neighbor gallery images.

3. Experiments

We evaluate our method on three publicly available datasets: FERET [21, 22], Multi-PIE [23, 24] and LFW-a [25, 26, 27, 28].

Face recognition across poses is conducted on FERET and Multi-PIE. During face recognition, each person has only one image with one specific pose angle as gallery while the remaining images of other poses form the probe set. Under this setting, we can calculate the accuracy for one specific pose angle in the probe set and also the average accuracy for all images in the probe set. In reporting experimental results, the average accuracy is denoted by Mean while the accuracy of one specific pose in the probe is referred as gallery-probe pair accuracy. It is worth to note that there is no overlapping person between feature learning and face recognition for the experiments on these two databases.

In addition, face recognition under unconstrained conditions is conducted on a subset of LFW-a, following the protocol from [29] and [30] strictly. Due to the limited person number and difficulties of unconstrained face recognition, the persons are used both for feature learning and face recognition as done in [29] and [30] on this dataset.

3.1. Databases

FERET: The pose variant b subset of FERET is used in our experiments. There are 1800 images from 200 persons in this pose variant subset. As shown in Fig. 1, each person has 9 images, each of which is captured under one of the nine pose angles: *ba* (0°), *bb* (+60°), *bc* (+40°), *bd* (+25°), *be* (+15°), *bf* (−15°), *bg* (−25°), *bh* (−40°) and *bi* (−60°). All the images are cropped according to the public annotations and using the code from the authors of [8], and then they are resized to 100×100 without any further preprocessing. We follow the evaluation protocol in [8], under which images of the first 100 persons are used for feature learning, and images of the remaining 100 subjects are regarded as the face recognition testing set. Therefore, there are 900 images for training and 900 images for testing. Based on the training images, 3600 pairs in S and 4950 pairs in D are constructed for feature learning.

Fig. 1. FERET: examples of face images from one person.

Fig. 2. Multi-PIE: examples of face images from one person.

Multi-PIE: Following the protocol in [14, 7, 31, 15], images under seven different poses are used in our experiments, which are 080(−45°), 130(−30°), 140(−15°), 051(0°), 050(+15°), 041(+30°) and 190(+45°) as shown in Fig. 2. There are 337 persons in Multi-PIE, and images of the first 200 subjects are employed for feature learning while images of the remaining 137 persons for face recognition testing, resulting in 1400 training and 959 testing images. All the images are cropped according to the public annotations and using the code from the authors of [8]. The cropped images are resized to 100×100 dimensions without any further preprocessing. For feature learning, S consists of 4200 image pairs while D is composed of 19900 pairs.

LFW-a: This database is widely used for face verification. To conduct face recognition experiments on it, we follow the protocol from [29] and [30], in which 86 people with 11 to 20 images are used. In total, there are 1251 images, among which 15 images of one person are shown as examples in Fig. 3. All the images are cropped from the center of the image and then resized to 32×32 dimensions for feature extraction. To report the result, 10-round face recognition experiments are conducted. At each round, 8 images from each person are randomly selected for training while the remaining ones are used for face recognition testing, i.e., 688 and 563 images in total for training and testing, respectively, at each round.

3.2. Features

Experiments are conducted based on two popular local features: SIFT [32] and LPQ [33], which have shown promising performance for face recognition in general [34, 35].

We apply Principal Component Analysis (PCA) to reduce the dimension of SIFT, LPQ and the self-similarity feature to the


Fig. 3. LFW-a: examples of face images from one person.

same number before feature learning. The PCA is trained using the same training data for feature learning. In the following experiments, different feature dimensions are tested. In addition to the subtracted feature, the original PCA processed SIFT and LPQ features are also used for experiments as baselines.

3.2.1. Feautures on FERET and Multi-PIE

SIFT: An image is segmented into 400 non-overlapping blocks, each of which has 5×5 pixels. In each block, a 128-dimension SIFT descriptor is extracted. After concatenating 400 SIFT descriptors, a 51200-dimension SIFT feature vector is generated. Since there are 400 blocks, a 79800-dimension self-similarity vector are obtained from a SIFT feature vector through equations 3 and 4.

LPQ: An image is divided into 100 non-overlapping blocks with the size of 10×10 pixels. A 256-dimension LPQ descriptor is extracted from each block. After concatenating them, a 25600-dimension LPQ feature is obtained. Through equations 3 and 4, a 4950-dimension self-similarity vector is generated based on the LPQ descriptors from 100 blocks.

3.2.2. Features on LFW-a

SIFT: An image is zero-padded to 35×35 dimensions and then segmented into 49 non-overlapping blocks, each of which consists of 5×5 pixels. In each block, a 128-dimension SIFT descriptor is extracted. After concatenating 49 SIFT descriptors, a 6272-dimension SIFT feature vector is generated. Based on the SIFT descriptors from 49 blocks, a 1176-dimension self-similarity vector are obtained through equations 3 and 4.

LPQ: An image is divided into 64 non-overlapping blocks with the size of 4×4 pixels. And then, a 256-dimension LPQ descriptor is extracted from each block. After concatenating them, a 16384-dimension LPQ feature is obtained. A 2016-dimension self-similarity vector is generated based on the LPQ descriptors from the 64 blocks through equations 3 and 4.

3.3. Results on FERET

To conduct the experiments on FERET, α is set to 10^{-4}. Apart from this, different feature types, feature dimensions and distance measures for self-similarity vectors are employed to evaluate the proposed method. Since there is only one image in the gallery for each peron under face recognition across poses, 1-NN method is used for the experiments on this database.

We first report the results using two different features (LPQ and SIFT) with 500-dimension and 2-norm distance based self-similarity vector. The mean accuracy results using different poses as gallery are listed in Table 1. Detailed gallery-probe pair results of SIFT feature can be found in Fig. 4. For both types of feature, as can be found in Table 1, our method is able to significantly improve the recognition performance over the original PCA processed feature, especially for the profile galleries, e.g., the mean accuracy of *bi* gallery is increased by 48.1% for the SIFT feature. As displayed in Fig. 4, our method successfully boosts the performance for every gallery-probe pair over the original PCA processed SIFT. Overall, the SIFT feature achieves better results than LPQ. One thing worth to mention is that the best result, 94.6%, is obtained using *be*

Table 1. Mean accuracies in percentage on FERET for different pose galleries using different features with 500 feature dimension and 2-norm distance based spatial self-similarity feature.

Gallery	LPQ		SIFT	
	PCA	Proposed	PCA	Proposed
bi	16.8	49.8	28.5	**76.6**
bh	31.0	70.5	45.6	**87.1**
bg	45.5	74.5	49.9	**89.9**
bf	46.3	78.9	52.6	**93.4**
ba	42.3	78.3	49.0	**91.8**
be	38.9	82.3	48.3	**94.6**
bd	33.5	77.5	39.1	**91.4**
bc	24.6	69.4	35.5	**84.9**
bb	19.0	47.8	26.5	**70.9**

Table 2. Mean accuracies in percentage of different pose galleries on FERET using different distance methods, 1-norm(l_1), square of 1-norm(l_1^2), 2-norm(l_2) and square of 2-norm(l_2^2) distance, for self-similarity feature. 500-dimension SIFT feature are used here.

Gallery	PCA	l_1	l_1^2	l_2	l_2^2
bi	28.5	**76.6**	73.3	76.6	74.9
bh	45.6	85.8	84.3	**87.1**	86.8
bg	49.9	89.5	87.5	**89.9**	89.6
bf	52.6	92.3	91.8	**93.4**	92.5
ba	49.0	90.5	89.5	**91.8**	90.9
be	48.3	92.9	92.3	**94.6**	93.8
bd	39.1	91.3	91.1	**91.4**	**91.4**
bc	35.5	**86.6**	85.9	84.9	86.1
bb	26.5	69.3	69.1	70.9	**73.3**

as gallery instead of the frontal image *ba*, of which the mean accuracy is 91.8%. One possible reason for this is that the pose angle of *be* is $+15°$, which is not large, and in this case, our method is able to maintain similar results for *bi* and *bh* while obtaining better results for *bc* and *bb* compared to using *ba* as gallery. This is also the case for the result of using *bf* ($-15°$) as gallery, of which the mean accuracy is 93.4%.

Furthermore, different PCA feature dimensions with 2-norm distance based self-similarity vector are tested using SIFT, of which results are displayed in Fig. 5. Our method is able to improve the performance under all feature dimensions from 100 to 500 compared to the PCA processed SIFT feature. Features with larger dimensions are able to preserve more face information, therefore, better results are obtained using larger feature dimensions as shown in Fig. 5.

In addition, we also test different distance measures for computing the spatial self-similarity feature using the 500-dimension SIFT feature. Specifically, 1-norm(l_1), square of 1-norm(l_1^2), 2-norm(l_2) and square of 2-norm(l_2^2) distance measures are used for this experiment. As can be seen in Table 2, our method works well using different distance measures. Although the results of different distances vary a bit for different pose galleries, the learned features all outperform the PCA processed SIFT with a large margin.

In the end, our method is compared with two similar state-of-the-art methods, DAE[36] and SPAE[15]. The results of the referenced methods are taken from [15], which also follow the

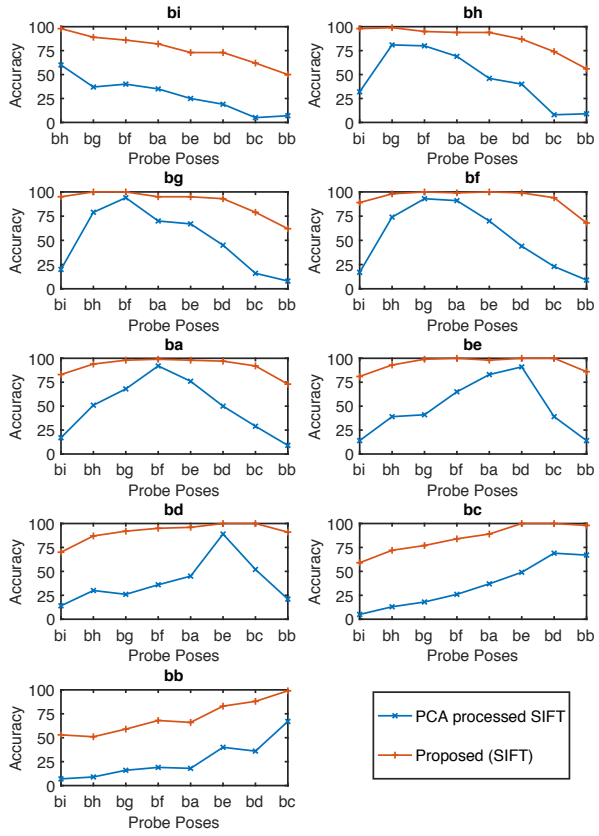


Fig. 4. Gallery-probe pair accuracies in percentage of PCA processed SIFT and the proposed method (SIFT) on FERET. The pose name on top of the figure is the pose used as gallery. 500-dimension SIFT feature and 2-norm based self-similarity feature are used.

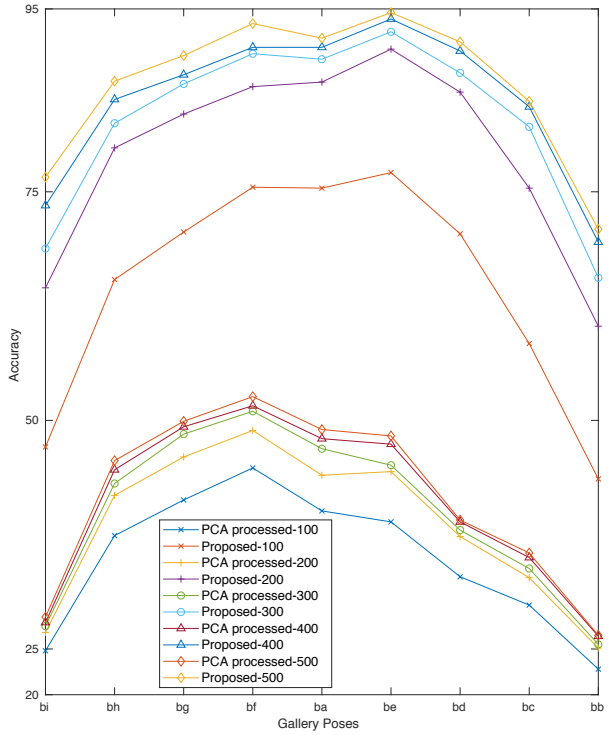


Fig. 5. Mean accuracies in percentage of different pose galleries on FERET using SIFT feature with different dimensions. A 2-norm distance based self-similarity feature is used.

Table 3. Comparison of accuracies in percentage with other state-of-the-art pose-free methods on FERET dataset using *ba* as gallery. The results of other methods are taken from [15].

Probe	DAE[36]	SPAE[15]	Proposed (SIFT)	
	ba			be
bi	61	77	**83**	81
bh	83	**95**	94	93
bg	94	**99**	98	99
bf	96	**99**	**99**	100
ba	–	–	–	98
be	96	**99**	98	–
bd	93	**98**	97	100
bc	91	**96**	92	100
bb	62	**77**	73	86
Mean	84.5	92.5	91.8	94.7

Table 4. Mean accuracies in percentage on Multi-PIE using different pose images as gallery with different feature type and dimensions and values of α.

Gallery	LPQ		SIFT	
	PCA	Proposed	PCA	Proposed
500, $\alpha = 10^{-4}$				
080	18.6	62.4	26.8	**78.0**
130	28.1	72.3	36.7	**87.0**
140	27.5	77.9	35.5	**92.0**
051	22.5	81.0	32.1	**95.1**
050	31.1	79.9	42.1	**91.1**
041	29.1	73.1	34.9	**86.0**
190	26.9	66.2	35.9	**83.5**
1000, $\alpha = 10^{-5}$				
080	19.3	63.3	27.5	**78.3**
130	30.0	73.1	37.1	**87.0**
140	28.7	82.0	36.4	**95.3**
051	24.1	86.1	33.3	**97.2**
050	34.2	82.4	42.9	**94.0**
041	30.2	74.5	35.4	**89.7**
190	27.9	66.4	36.7	**84.1**

same protocol as we do, but only the results using *ba* as gallery are provided. Both compared methods are based on deep learning. As listed in Table 3, our method achieves the second best mean accuracy which is close to the best one using *ba* as gallery. It is worth to note that our method obtains better result using *be* as gallery as compared with using *ba* as gallery. Moreover, the method in [15] requires labelled frontal face images to train its model while our method does not.

3.4. Results on Multi-PIE

For all the experiments on Multi-PIE, 1-NN method is used for face recognition across poses since there is only one image for each person in the gallery. Beside this, 2-norm distance based self-similarity feature is used. Since there are more training data in Multi-PIE, 1000-dimension features are tested in addition to the 500-dimension features. Furthermore, we set $\alpha = 10^{-5}$ for 1000 dimension feature for that much more training pairs exist in D. The mean accuracies of different pose gal-

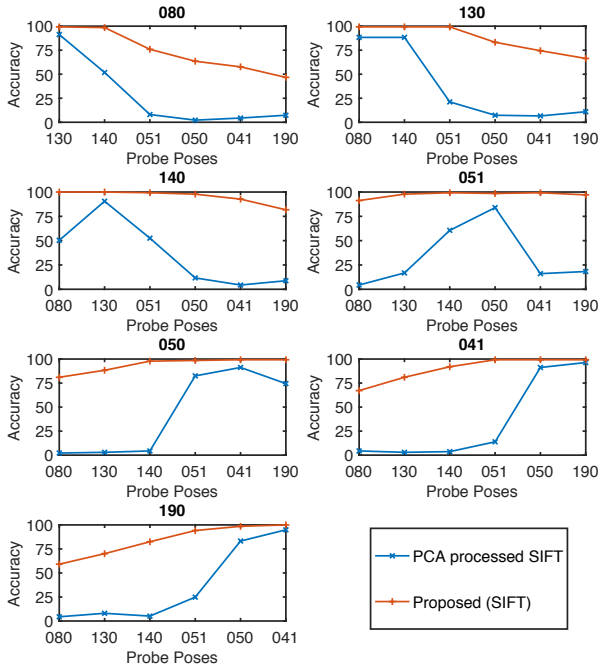


Fig. 6. Gallery-probe pair accuracy results in percentage of PCA processed SIFT and the proposed method (SIFT) on Multi-PIE. The pose name on top of each figure is the pose used as gallery. 1000-dimension, 2-norm distance based self-similarity feature and $\alpha = 10^{-5}$ are used.

leries can be found in Table 4. Detailed gallery-probe accuracy results using the SIFT feature with 1000 dimensions and $\alpha = 10^{-5}$ are illustrated in Fig. 6. Overall, the SIFT feature achieves better results than LPQ, and features with larger dimensions obtain better results as well. Compared to the PCA processed feature, our method successfully improves the performance with a large margin for the mean accuracies as shown in Table 4. The accuracy of every gallery-probe pair is also successfully increased as displayed in Fig. 6. The best mean accuracy, 97.2%, is achieved by using frontal image as gallery and the 1000-dimension SIFT feature with a 63.8% increase over the PCA processed feature.

We also compare our method with other state-of-the-art pose-free methods. The frontal image (051) is used as gallery. The same testing protocol in [8] is followed to make comparison. As shown in Table 5, our method achieves the second best mean accuracy. In [14], the FIP method uses one middle layer as feature and RL treats the reconstructed frontal image as feature. The mean accuracy of our method is close to that of RL, and better than that of FIP. In addition, the method in [14] requires labelled frontal face images to train its model while our method does not. Besides theses, RL [14], FIP [14] and SPAE [15] are all based on deep learning, which usually require a larger number of multi-pose training images [1].

3.5. Results on LFW-a

Since our method does not need any labelled pose for face images which is opposite to the methods in [14, 15] where labelled frontal face images are needed, it is possible to conduct face recognition under unconstrained conditions on LFW-a using our method. As there are eight images in the gallery for each person, 8-NN based face recognition are conducted on LFW-a. The feature dimension is reduced to 100 for that there are less

Table 5. Comparison with other state-of-the-art pose-free methods on Multi-PIE. The results of the other methods are taken from [1] or the original papers. Accuracies are in percentage. Frontal images (051) are used as gallery.

Probe	RL [14]	FIP [14]	MRFs [7]	RFG [31]	SPAE [15]	Proposed SIFT
080	**95.6**	93.4	86.3	82.9	84.9	91.2
130	**98.5**	95.6	89.7	88.3	92.6	97.8
140	**100.0**	100.0	91.7	93.2	96.3	99.3
050	**99.3**	98.5	91.0	93.4	95.7	98.5
041	98.5	96.4	89.0	91.6	94.3	**99.3**
190	**97.8**	89.8	85.7	85.5	84.4	97.1
Mean	**98.3**	95.6	88.9	89.2	91.4	97.2

Table 6. Mean accuracie in percentage with stadard deviation of 10-random rounds of experiments on LFW-a.

		$\alpha = 10^{-1}$	$\alpha = 10^{-2}$	$\alpha = 10^{-4}$
LPQ	PCA	38.2 ± 1.9		
	Proposed	**49.2** ± 1.2	44.9 ± 1.4	44.2 ± 1.5
SIFT	PCA	23.7 ± 0.9		
	Proposed	36.1 ± 2.2	31.6 ± 1.9	31.1 ± 1.8
SRC[30]		38.1 ± 0.011		
LCLE-DL[30]		38.8 ± 0.009		

training images in this dataset. We follow the testing protocol in [30] to report our results. As shown in Table 6, we vary α among 10^{-1}, 10^{-2} and 10^{-4}, under all of which our method successfully improved the performance compared to the original PCA processed feature. Using LPQ feature, our method achieves better result than the other state-of-the-art methods under all different values of α, which shows that our method is able to work under unconstrained situations as well. As can be found in Table 6, the best result is 49.2% achieved by the proposed method using LPQ with 10^{-1}.

3.6. Complexity Analysis

During training procedure, as can be found in equation 17, the complexity of our method is dependent on the number of training samples and feature dimension besides the feature extraction and PCA training procedure. For face recognition, there exist two extra operations than using the PCA processed local feature directly, self-similarity feature extraction and linear transformation using equation 5.

The running time of the proposed method and original PCA processed feature on LFW-a can be found in Table 7. All the experiments are conducted on a desktop computer with i7-6700K (4.0GHz) CPU and 64GB memory. The code [1] is implemented using Python without any optimization especially for the self-similarity feature extraction. As shown in Table 7, the training of our method is quite fast, which makes our method applicable to online applications, and the recognition of our method is fast enough for real-time application. Moreover, since feature extraction occupies a large proportion of the processing time, it is possible to further accelerate our method using optimized algorithms of feature extraction [37].

[1] http://kom.aau.dk/~zt/online/FeatureLearning_FaceRec_PRL.zip

Table 7. Runing time in seconds on LFW-a. The training time includes feature extraction, PCA training and transformation, and feature learning, of which time is listed in the parentheses, respectively. The testing time is for one image including feature extraction, PCA transformation and subtraction, while the time in the parentheses is for the original PCA processed feature.

	LPQ	SIFT
Training	11.60 (9.41, 2.17, 0.02)	4.08 (3.27, 0.79, 0.02)
Testing	0.02197 (0.01471)	0.01414 (0.00863)

4. Conclusion

In this paper, we proposed a feature learning method based on the spatial self-similarity vector for varying-pose face recognition without the need of prior pose knowledge. The problem caused by pose variation is handled by subtracting the pose related part from a local feature instead of converting a profile view to a frontal face. This is done by learning a linear transformation of the self-similarity feature through minimizing the distance between subtracted feature vectors from the same person under different poses and retaining the discriminative capability of the subtracted feature at the same time. Using the proposed feature learning method, the performance of face recognition across poses and even under unconstrained conditions is improved significantly over the original feature. Compared to existing state-of-the-art methods, our method obtains better or comparable results while at the same time, having a low-complexity.

References

[1] C. Ding, D. Tao, A comprehensive survey on pose-invariant face recognition, ACM Transactions on Intelligent Systems and Technology (TIST) 7 (2016) 37.

[2] Y. Gao, H. J. Lee, Cross-pose face recognition based on multiple virtual views and alignment error, Pattern Recognition Letters 65 (2015) 170–176.

[3] A. K. Jain, S. Z. Li, Handbook of face recognition, Springer, 2005.

[4] W. Zhao, R. Chellappa, P. J. Phillips, A. Rosenfeld, Face recognition: A literature survey, ACM Computing Surveys (CSUR) 35 (2003) 399–458.

[5] X. Zhang, Y. Gao, Face recognition across pose: A review, Pattern Recognition 42 (2009) 2876–2896.

[6] X. Chai, S. Shan, X. Chen, W. Gao, Locally linear regression for pose-invariant face recognition, Image Processing, IEEE Transactions on 16 (2007) 1716–1725.

[7] H. T. Ho, R. Chellappa, Pose-invariant face recognition using markov random fields, Image Processing, IEEE Transactions on 22 (2013) 1573–1584.

[8] A. Sharma, D. W. Jacobs, Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 593–600.

[9] A. Sharma, M. A. Haj, J. Choi, L. S. Davis, D. W. Jacobs, Robust pose invariant face recognition using coupled latent space discriminant analysis, Computer Vision and Image Understanding 116 (2012) 1095–1110.

[10] A. Li, S. Shan, W. Gao, Coupled bias–variance tradeoff for cross-pose face recognition, Image Processing, IEEE Transactions on 21 (2012) 305–315.

[11] M. Sonka, V. Hlavac, R. Boyle, et al., Image processing, analysis, and machine vision, volume 3, Thomson Toronto, 2008.

[12] G. Passalis, P. Perakis, T. Theoharis, I. A. Kakadiaris, Using facial symmetry to handle pose variations in real-world 3d face recognition, Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (2011) 1938–1951.

[13] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT Press, 2016.

[14] Z. Zhu, P. Luo, X. Wang, X. Tang, Deep learning identity-preserving face space, in: Computer Vision (ICCV), 2013 IEEE International Conference on, 2013, pp. 113–120.

[15] M. Kan, S. Shan, H. Chang, X. Chen, Stacked progressive auto-encoders (spae) for face recognition across poses, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, 2014, pp. 1883–1890.

[16] O. M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition., in: BMVC, volume 1, 2015, p. 6.

[17] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A discriminative feature learning approach for deep face recognition, in: ECCV, Springer, 2016, pp. 499–515.

[18] X. Duan, Z.-H. Tan, Local feature learning for face recognition under varying poses, in: Image Processing (ICIP), 2015 IEEE International Conference on, IEEE, 2015, pp. 2905–2909.

[19] B. Kulis, Metric learning: A survey, Foundations & Trends in Machine Learning 5 (2012) 287–364.

[20] A. Bellet, A. Habrard, M. Sebban, A survey on metric learning for feature vectors and structured data, arXiv preprint arXiv:1306.6709 (2013).

[21] P. J. Phillips, H. Wechsler, J. Huang, P. J. Rauss, The feret database and evaluation procedure for face-recognition algorithms, Image and Vision Computing 16 (1998) 295–306.

[22] P. J. Phillips, H. Moon, S. A. Rizvi, P. J. Rauss, The feret evaluation methodology for face-recognition algorithms, Pattern Analysis and Machine Intelligence, IEEE Transactions on 22 (2000) 1090–1104.

[23] R. Gross, I. Matthews, J. Cohn, T. Kanade, S. Baker, Multi-pie, in: Automatic Face Gesture Recognition, 2008 IEEE International Conference on, 2008, pp. 1–8.

[24] R. Gross, I. Matthews, J. Cohn, T. Kanade, S. Baker, Multi-pie, Image and Vision Computing 28 (2010) 807–813.

[25] G. B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[26] G. B. H. E. Learned-Miller, Labeled Faces in the Wild: Updates and New Reporting Procedures, Technical Report UM-CS-2014-003, University of Massachusetts, Amherst, 2014.

[27] L. Wolf, T. Hassner, Y. Taigman, Effective unconstrained face recognition by combining multiple descriptors and learned background statistics, Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (2011) 1978–1990.

[28] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, G. Hua, Labeled faces in the wild: A survey, in: Advances in face detection and facial image analysis, Springer, 2016, pp. 189–248.

[29] S.-J. Wang, J. Yang, M.-F. Sun, X.-J. Peng, M.-M. Sun, C.-G. Zhou, Sparse tensor discriminant color space for face verification, Neural Networks and Learning Systems, IEEE Transactions on 23 (2012) 876–888.

[30] Z. Li, Z. Lai, Y. Xu, J. Yang, D. Zhang, A locality-constrained and label embedding dictionary learning algorithm for image classification, Neural Networks and Learning Systems, IEEE Transactions on 28 (2017) 278–293.

[31] M. Kafai, L. An, B. Bhanu, Reference face graph for face recognition, Information Forensics and Security, IEEE Transactions on 9 (2014) 2132–2143.

[32] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2004) 91–110.

[33] T. Ahonen, E. Rahtu, V. Ojansivu, J. Heikkila, Recognition of blurred faces using local phase quantization, in: Pattern Recognition, 2008 International Conference on, IEEE, 2008, pp. 1–4.

[34] B. Yuan, H. Cao, J. Chu, Combining local binary pattern and local phase quantization for face recognition, in: Biometrics and Security Technologies (ISBAST), 2012 International Symposium on, IEEE, 2012, pp. 51–53.

[35] C. Geng, X. Jiang, Sift features for face recognition, in: Computer Science and Information Technology, 2009 IEEE International Conference on, IEEE, 2009, pp. 598–602.

[36] Y. Bengio, et al., Learning deep architectures for ai, Foundations and trends® in Machine Learning 2 (2009) 1–127.

[37] L.-C. Chiu, T.-S. Chang, J.-Y. Chen, N. Y.-C. Chang, Fast sift design for real-time visual feature extraction, Image Processing, IEEE Transactions on 22 (2013) 3158–3167.