Locally Kernel-based Nonlinear Regression for Face Recognition

Yaser Arianpour[1], Sedigheh Ghofrani[1] and Hamidreza Amindavar[2]

[1]*Islamic Azad University,*
South Tehran Branch, Electrical Engineering Department, Tehran, Iran

[2]*Amirkabir University of Technology,*
Electrical Engineering Department,Tehran, Iran

st_y_arianpour@azad.ac.ir, s_ghofrani@azad.ac.ir and hamidami@aut.ac.ir

Abstract

The variation of facial appearance due to the viewpoint or pose obviously degrades the accuracy of any face recognition systems. One solution is generating the virtual frontal view from any given non-frontal view to obtain a virtual gallery/probe face. As the state-of-the-art face recognition algorithm, linear regression computes a reconstruction matrix from the images of each subject and then approximates the probe face image by using the reconstruction matrix, but the performance of this linear algorithm is limited due to the nonlinear structure of the face images which is caused by variations in illumination, expression, pose and occlusion. Following this idea, in this paper, we propose an efficient and novel locally kernel-based nonlinear regression (LKNR) method, which generates the virtual frontal view from a given non-frontal face image. Because of the high (even infinite) dimensionality of the nonlinear transformation functions, it is infeasible to directly calculate the corresponding reconstruction matrix and therefore is unable to approximate explicitly the probe image. So, with the help of kernel functions, we overcome to this mentioned problem by embedding the nonlinear regression in the stage of computing the reconstruction matrix from the non-frontal input face and non-frontal face database. The comparison of the proposed method with locally linear regression (LLR) and eigen light-field (ELF) methods is also provided in terms of the face recognition accuracy. Experimental results show that the proposed method outperforms two other methods in terms of robustness and visual effects.

***Keywords**: Face recognition, kernel function, locally kernel-based nonlinear regression (LKNR), reconstruction matrix, virtual frontal view*

1. Introduction

Face recognition has been studied for more than three decades. The state-of-the-art recognition technologies can achieve appropriate accuracy under predefined cases, such as frontal faces images when the lighting is controlled [3-5]. However, the most current face recognition systems are pretty sensitive to pose, lighting, occlusion and aging. It means they fail under uncontrolled conditions such as outdoor captured images with uncooperative subjects. Among those above mentioned, pose is the bottlenecks because of varying appearance of a person with different pose. Therefore, the typical appearance based methods, such as eigenface [6], degrade dramatically when non-frontal probes match against the enrolled frontal faces.

Many approaches have been proposed to deal with pose problem, and the view-based methods are widely used [7-11]. The drawback of using the view-based method is that it usually needs multiple face images with different poses for each subject. Gross, et al., [12-13] proposed the eigen light-field (ELF) method to tackle the pose problem. This approach also needs an extra independent training set (differs of gallery) that contains multiple images of different poses for each subject.

Generating virtual view is another possible solution for pose invariant face recognition. By generating virtual view, one can either the normalize the all face images to a predefined pose (e.g., frontal) or expand the gallery (or the training set) to cover the large pose variations. Simply speaking, there are two strategies to generate the virtual view: 3-D model-based method [14-20] and learning-based method [21-25].

Since the variations in appearance caused by pose are closely related to the 3-D face structure, it is a natural idea to recover the 3-D model from the input 2-D face image. Thus, virtual views under any viewpoint are easily generated by using graphic rendering techniques [14, 15, 18]. The 3-D Morphable Model [14-15] is one successful technology for recovering the 3-D face model. In this method, the prior knowledge of the face shape and texture is modeled by principal component analysis (PCA). Then any new face can be modeled by the linear combination of the prototypes, in which the corresponding shape and texture are expressed by the exemplar faces respectively. The specific 3-D face can be recovered automatically from one or more photographs by simultaneously optimizing the shape, texture and mapping parameters through an analysis-by-synthesis strategy. However, it is time consuming for most real-world applications. To reduce the complexity, Jiang, et al., [20] proposed a simplified version of 3-D morphable model to reconstruct the specific 3-D face from a frontal face. Lee [19] also realized the 3-D deformable model for the 3-D face reconstruction which was composed of the edge, color region and wire frame models. Generic 3-D face model has been used in many papers to generate the virtual views to tackle the pose problem, such as [16] and [17]. The illumination Cone method [18] can also reconstruct the accurate shape and albedo for a specific person from at least seven images under a fixed pose but with different lighting conditions.

Unlike 3-D model-based methods, learning-based approaches generally try to learn how to estimate a virtual view directly in 2-D domain [21-25]. Beymer and Poggio proposed an example-based algorithm to synthesize novel views from single image and applied them for face recognition [22-23]. Vetter, et al., [24-25] proposed a method, in which a face image was separated into shape vector and texture vector, and the linear object classes [21] were applied to generate the virtual shape and texture under a novel pose. Then the virtual "rotated" images are generated easily by combining the generated shape and texture. Evidently, the quality of the novel virtual view heavily depends on the accuracy of the face alignment (i.e., the separation of shape and texture).

Chai, et al., [2] proposed the locally linear regression (LLR) method to efficiently generate the virtual frontal view from a given non-frontal face image. They partitioned the non-frontal face image into multiple patches and applied linear regression to each patch to predict its corresponding frontal patch. In comparison with [25], LLR is more efficient because only simple linear regression is needed. In addition, LLR requires only the centers of the two eyes for alignment rather than accurate face alignment, which is mandatory for the linear object classes (LOC) method [21]. However, the performance of this linear algorithm is limited due to the nonlinear structure of the face images which is caused by variations in illumination, expression, pose and occlusion.

In this paper, we propose a nonlinear regression method, which is based on kernel solutions. Although the linear assumption could dramatically decrease the computational consumption, but also eliminate lots of information since the rotation of a human head is known as a nonlinear problem, therefore, we propose a kernel-based nonlinear regression algorithm for generating virtual frontal view from any given non-frontal view to obtain both a virtual gallery/probe face and effective face recognition. Because of the high (even infinite) dimensionality of the nonlinear transformation functions, it is infeasible to directly calculate the corresponding reconstruction matrix and thus, it is unable to explicitly approximate the virtual frontal view. For this purpose, with the help of kernel functions, we overcome to the computational complexity of high dimensional nonlinear transformation functions by embedding these functions in the stage of computing the reconstruction matrix from the non-frontal input face and non-frontal face database.

This paper is organized as follows. In Section 2, the linear regression is explained. In Section 3, the global and local nonlinear regression based on kernel functions are expressed. The experimental results of our proposed algorithm are provided in Section 4. Finally, in Section 5, we have conclusion.

2. Linear Regression

Given a non-frontal facial image, this is to generate its virtual frontal view based on a training set. Chai *et al.* [2] formulated this problem mathematically as a regression task. They considered the training set as $\{(X_0, X_P)\}$, where $X_0 = (I_0^1, I_0^2, ..., I_0^N)$ and $X_P = (I_P^1, I_P^2, ..., I_P^N)$ denote the frontal face set and non-frontal face set all with pose P belong to N subjects. It is to notify that both I_P^i and I_0^i correspond to the same person but with different pose. The linear regression method [2] implies an specific transform in order to convert any non-frontal face I_P into its frontal counterpart I_0, the mapping is written as follows:

$$I_0 = A\,I_P, \tag{1}$$

Let n denotes the number of pixels of an image, in case $n > N$, also the linear operator A is defined as $A = X_0\,(X_P)^{\perp}$, where $(X_P)^{\perp} = \left((X_P)^T\,(X_P)\right)^{-1}(X_P)^T$ is the pseudo inverse of X_P. Once the linear operator A is already estimated based on the training set, when given any image I_P with pose P, its corresponding virtual frontal image I_0 can be computed according to the same linear transformation, i.e.

$$I_0 = A\,I_P = X_0\,(X_P)^{\perp}\,I_P. \tag{2}$$

Considering the virtual view generation in a step further, Chai *et al.* [2] rewrote the equation (2) as:

$$I_0 = X_0\,\alpha, \tag{3}$$

where $\alpha = (X_P)^{\perp}\,I_P$ is named the reconstruction coefficients. Therefore, the reconstruction coefficients of pose P are to be obtained at first for virtual view generation, the procedure is sketched in Fig. 1. Actually, the above solution of the reconstruction coefficients is the result of an optimization procedure aiming at seeking a coefficients vector, which can best represent the input image in the P pose image space. This is achieved by minimizing the following residue function:

$$\varepsilon(\alpha) = \left\| I_P - I_{P,\mathrm{Rec}} \right\|^2, \tag{4}$$

where $I_{P,\mathrm{Rec}} = X_P\,\alpha = \sum_{j=1}^{N} I_{P,j}\,\alpha_j$ is the projection of I_P in the P pose image space. In addition, from the above analysis, one can understand the linear regression more clearly as follows: the virtual frontal view of an input non-frontal face image I_P with pose P is generated through a linear combination by using the same coefficients reconstructing the I_P in the P pose image space (Figure 1). Coincidentally, this idea is consistent with the concept of linear object class [24], but Chai, et al., [2] formulated the problem differently from the point of view of regression.

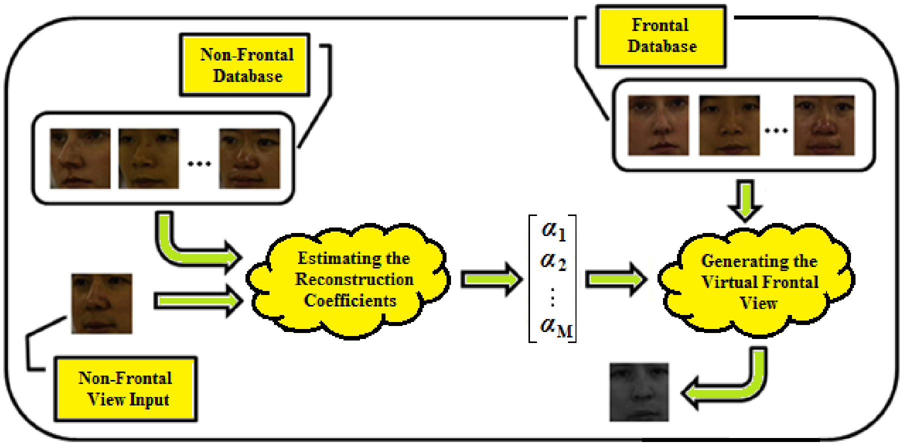


Figure 1. The Block Diagram of Generating the Virtual Frontal View based on Using the Linear Regression Method

3. Nonlinear Regression

The main advantage of using the linear regression in order to generate the virtual frontal view is easy implementation and the main drawback is unrobust to variations such as viewpoints, illumination or expression. In this paper, in order to overcome to this problem, we propose the local nonlinear regression based on kernel method. In following, at first the kernel method and three type kernels are explained and then our proposed method, locally kernel-based nonlinear regression (LKNR) is explained.

3.1 Kernel Functions

The linear assumption could dramatically decrease the computational consumption, but also eliminate lots of information since the rotation of a human head is known as a nonlinear problem. In appearance-based face recognition, many nonlinear methods have been proposed, and among them, kernel-based methods are very effective and have been proved that they can extract nonlinear features providing better recognition results. Based on Cover's theorem [26], patterns which are separated nonlinearly in an input space may linearly be separated if the input space is transformed nonlinearly into a high-dimensional feature space. In kernel-based methods, there is a nonlinear mapping Ψ which maps the input data space $\Re^n$ to the feature space F. Suppose that an input vector $x_i \in \Re^n$ belongs to the original space. This vector will be mapped into a potentially higher dimensional vector:

$$\begin{aligned} \Psi : \Re^n &\rightarrow \mathrm{F} \\ x &\rightarrow \Psi(x). \end{aligned} \tag{5}$$

Instead of specifically clarifying the nonlinear mapping function Ψ, the inner product relationship between vector pairs in feature space is defined:

$$\kappa(x_i , x_j) = \langle \Psi(x_i) , \Psi(x_j) \rangle = \Psi(x_i)^T \ \Psi(x_j) , \tag{6}$$

Obviously, it is hard to realize matrix operations in an extremely high dimensional space. However, we could accomplish the calculation indirectly. The reconstruction coefficients of equation (3) ensure the calculation could be simplified even though the dimension of mapped vector is extremely high [27]. In general, there are four important and practical classes of kernel functions [28] in pattern recognition and image processing applications, which named linear, polynomial, Gaussian and sigmoid kernels. In this paper, we choose three common kernels which are explained in following.

For any x_i , x_j belong to the input space, the linear kernel [28] is:

$$\kappa(x_i , x_j) = x_i^T \ x_j + c \, , \tag{7}$$

where c is an optional constant. The Polynomial kernel [28] is:

$$\kappa(x_i , x_j) = (\beta \, x_i^T \ x_j + c)^d \ ; \ d = 1,2,...,L \, . \tag{8}$$

Adjustable parameters are the slope β , constant c and polynomial degree d. The Polynomial kernel is well suited where the training data has been normalized. The well known Gaussian kernel [28] is:

$$\kappa(x_i , x_j) = \exp\left(-\frac{\left\| x_i - x_j \right\|^2}{2\,\sigma^2} \right) . \tag{9}$$

where parameter σ refers to the standard deviation and play an important role. If it is overestimated, the exponential will behave almost linearly and the higher-dimensional projection lose its nonlinear power and if it is underestimated, the function will lack regularization and the decision boundary will be sensitive to noise for training data.

3.2. Nonlinear Regression with Kernel Functions

In this section, we use the same marks as them have been represented in Section 2. Let Ψ be a specific transform function, which map the n-dimensional vector to a high dimensional space. The kernel-based nonlinear regression algorithm for virtual frontal view generation is described as follows: At first, from the expression of predicted vector I_0 in the former section (i.e., equation (2)), we could straightforward get the expression of I_0 in the kernel space:

$$I_0 = A \, \Psi(I_P) \, , \tag{10}$$

where $A = X_0 \ \Psi(X_P)^{\perp}$ is a nonlinear operator, and the kernel mapping of matrix X_P is $\Psi(X_P) = \left[\Psi(I_P{}^1) \quad \Psi(I_P{}^2) \quad ... \quad \Psi(I_P{}^N) \right]$. With similar linear regression procedure, we can get the pseudo-inverse of $\Psi(X_P)$ as follow:

$$\begin{aligned} \Psi(X_P)^{\perp} &= \left(\Psi(X_P)^T \ \Psi(X_P) \right)^{-1} \Psi(X_P)^T \\ &= R^{-1} \ \Psi(X_P)^T \, , \end{aligned} \tag{11}$$

where $R = \Psi(X_P)^T \ \Psi(X_P)$ defines a $N \times N$ Gram matrix, and the elements of R can be determined by virtue of the kernel function :

$$R_{ij} = \left\langle \Psi(I_P{}^i) \, , \Psi(I_P{}^j) \right\rangle = \kappa(I_P{}^i \, , I_P{}^j) \, ; \; i, j = 1, 2, \dots, N \, . \tag{12}$$

In this way, the virtual frontal view can be derived by:

$$\begin{aligned} I_0 &= \left[X_0 \; \Psi(X_P)^{\perp} \right] \Psi(I_P) \\ &= X_0 \; R^{-1} \left[\Psi(X_P)^T \; \Psi(I_P) \right] \\ &= X_0 \; R^{-1} \, S \, , \end{aligned} \tag{13}$$

where also $S = \Psi(X_P)^T \; \Psi(I_P)$ defines a $N \times 1$ Gram matrix, and the elements of S can be determined by virtue of the kernel function :

$$\begin{aligned} S_k &= \left\langle \Psi(I_P{}^k) \, , \Psi(I_P) \right\rangle = \left[\kappa(I_P{}^1 \, , I_P) \quad \kappa(I_P{}^2 \, , I_P) \quad \dots \quad \kappa(I_P{}^N \, , I_P) \right] \\ &= \kappa(I_P{}^k \, , I_P) \, ; \; k = 1, 2, \dots, N \, . \end{aligned} \tag{14}$$

3.2.1 Global Kernel-based Nonlinear Regression: Given the above analysis, the virtual frontal view from the single non-frontal facial image can be easily derived by using equation (13). Note that, when this procedure is implemented, one should carefully align the face images. As it is well accepted in face recognition area, one can simply just align the faces according to the eye centers, and then the normalized face images in a whole are used to be feed into the above prediction. This implementation is called as global kernel-based nonlinear regression (GKNR). However the face is not planar in whole, that is to say the absolute linear mapping between two different views of a person does not exist, and therefore the nonlinear methods should be used. Since we use the kernel functions, this problem is partially solved and we get better results than the global linear regression introduced by Chai, et al., [2]. Nevertheless, still due to applying the entire surface of a face, some important areas of the face such as eyes, nose and mouth may not be recovered perfectly and so the generating virtual view in these areas will be blur (see Figure 3). Therefore, both the reconstruction of the input image in pose image space and the prediction in the frontal image space are not as precise as expected. Considering that some facial patch is more like a planar surface, a natural improvement of GKNR is applying nonlinear regression locally.

3.2.2 Local Kernel-based Nonlinear Regression: Based on the above problems that arise due to the intrinsic non-planar structure of the face and applying the entire surface, given that a 3-D face surface is composed of many local planar regions, for each small patch, the nonlinear mapping will be maintained better both in the single pose image space and across different poses than the global case. So we propose a method to synthesize virtual views by local kernel-based nonlinear regression (LKNR), in which nonlinear regression is conducted in patch-wise mode. Concretely, face images are partitioned into uniformed blocks, and then each block are predicted using nonlinear regression, as illustrated in Figure 2. This procedure is formally formulated as follows: Firstly, given the training set, each face image should be partitioned into M blocks. Due to the pose variation, different modes of partitioning images are performed according to their pose categories. In this method, the frontal faces are partitioned into regular grids, while the partitioning of images with P pose is completed by coarsely seeking for the counterpart of the frontal patches by the aid of an average 3-D face model. This ensures the corresponding local patches in frontal and pose image possess the same semantics, as can be seen in Figure 2.

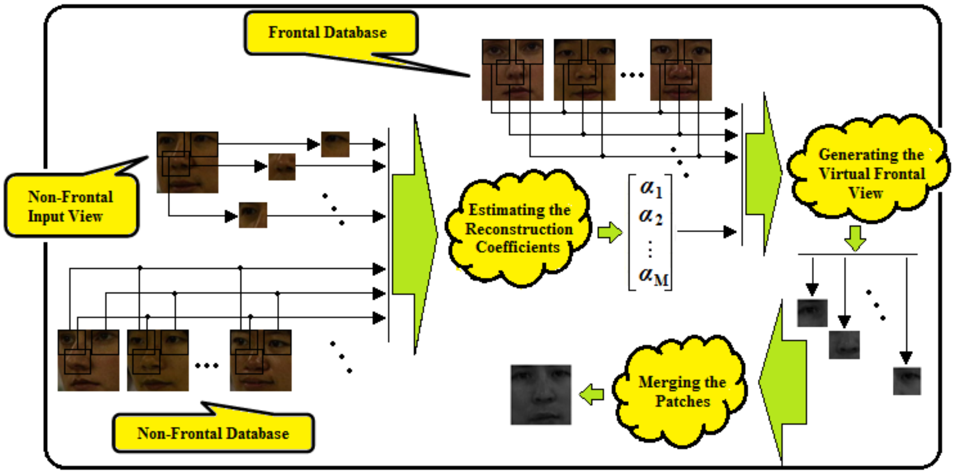


Figure 2. The Block Diagram of Generating the Virtual Frontal View based on using the LKNR Method

There by, given an input image I_P whose pose is P, they partition it into M small patches $I_P = \left(I_{(1,P)} \quad I_{(2,P)} \quad \dots \quad I_{(M,P)} \right)$ as is done for the P pose training images. Predicting the corresponding i-th frontal patch $I_{(i,0)}$ for the i-th non-frontal patch $I_{(i,P)}$ follows two steps:
Estimating the reconstruction coefficients for the i-th small input patch in the specific patch space by:

$$\alpha_i = R_i^{-1} S_i , \tag{15}$$

where $R_i = \Psi(X_{(i,P)})^T \, \Psi(X_{(i,P)})$ and $S_i = \Psi(X_{(i,P)})^T \, \Psi(I_{(i,P)})$; $i = 1, 2, \dots, M$ are Gram matrices and also $X_{(i,P)} = \left(I^1_{(i,P)} \quad I^2_{(i,P)} \quad \dots \quad I^N_{(i,P)} \right)$ is the i-th patches with P pose from the training set.
Obtaining the virtual frontal patch by:

$$I_{(i,0)} = X_{(i,0)} \, \alpha_i , \tag{16}$$

where $X_{(i,0)} = \left(I^1_{(i,0)} \quad I^2_{(i,0)} \quad \dots \quad I^N_{(i,0)} \right)$ is the i-th patch from the frontal images in the training set.
After performing such prediction for each patch in the I_P, all the small virtual frontal patches are combined into a whole vector, that is the target virtual frontal view $I_0 = \left(I_{(1,0)} \quad I_{(2,0)} \quad \dots \quad I_{(M,0)} \right)$. The resulting normalized frontal view can then be used for recognition.

4. Experimental Results

In this paper, we use the CMU PIE [29] and NCKU CSIE [30] face database in order to show the performance of our proposed method. The CMU PIE database includes 41,368 face images belong to 68 persons with 13 different poses, 43 different illumination conditions, and 4 different expressions. We use five pose subsets of CMU PIE database, which are pose sets 37 and 11 (i.e. yawing about ±45 degree), pose sets 05 and 29 (i.e. yawing about ±22.5 degree), and pose set 27 (nearly frontal face). The NCKU CSIE database includes 6660 face images belong to 90 persons with 37 different real poses and 37 different synthesized poses.

In this work, we use three pose subsets of NCKU CSIE database, which are the natural frontal faces and the poses yawing about ±40 degrees.

4.1 Generating the Virtual Frontal Face

In this work, the all face images are normalized to the same size with 160×160 pixels after fixing the eyes positions and keeping the aspects of face, as shown in Fig. 3, where Fig. 3-a and Fig. 3-c are the input non-frontal views from PIE P11 and PIE P29 subsets respectively. Fig. 3-b and Fig. 3-d illustrate the prediction results of GKNR for Fig. 3-a and Fig. 3-c respectively, i.e., using the whole 160×160 patch for prediction, and Fig. 3-e is its ground truth frontal views from PIE P27 subset. Two parameters are to be still obtained, i.e. the size of patches and the sampling step for a patch, though the face size is fixed. In the following experiments, in spite of using the linear kernel function is simpler than others, we use the polynomial kernel functions, due to controlling the contrast of produced image by varying the polynomial degree d. In this paper the polynomial degree d is considered to have a value between 0 and 1 for different input images. In addition, in order to reduce the computational cost, we do not use the Gaussian kernel function.

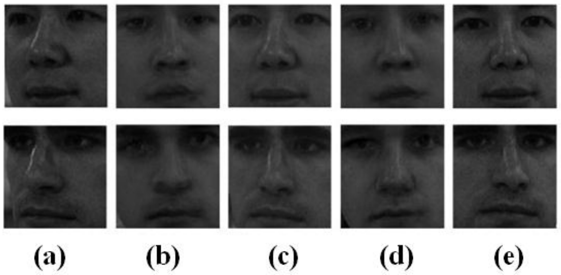


Figure 3. The Normalized Non-frontal Images belong to PIE P11 (a) and PIE P29 (c) Subsets. The virtual frontal view, (b) and (d), that generated by using GKNR from (a) and (c) respectively. (e) The normalized ground truth frontal image from PIE P27

Table 1. Example Results and the Corresponding PSNR (dB) of LKNR with Various Patch Size and Step

Patch Size	Step				
	5	10	20	40	80
20×20	 30.97	 31.41	 19.30		
40×40	 32.43	 32.79	 33.21	 19.48	
80×80	 32.20	 32.59	 32.97	 33.30	 20.44

To evaluate the prediction accuracy, the PSNR is calculated by using the following equation:

$$PSNR = 10 \times \log_{10} \frac{255^2}{\frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}\left[f(i,j) - \hat{f}(i,j) \right]^2}, \tag{17}$$

where $f(i,j)$ is the ground truth frontal view, $\hat{f}(i,j)$ is the predicted virtual frontal view, and W and H are the width and height of the image respectively. According to our experimental results which are shown in Table 1, the patch size is to be considered neither too large nor too small. A too large patch may cause the break of linear assumption in local linear regression although the proposed method makes use of kernel functions and a too small patch may result in serious mis-alignment since we use a generic 3-D model. Therefore, a small patch size may result in more artifacts, especially for patch size 20×20 and the large patch size may result in more blurring effect, especially for the nose and mouth parts. As for the sampling step, it is a tradeoff between over smoothing and blocking effect. Especially, more blocking effects can be observed, when the patches are sampled without any overlapping, i.e., the step is as large as the patch size. On the contrary, a small step such as 1 pixel always results in very smoothing face images.

The experimental results for two methods, GKNR and LKNR, are shown in Figure 3 and Table 1, where the three patch sizes (20×20, 40×40 and 80×80) and five sampling steps (5, 10, 20, 40 and 80) are being used. According to our achievements, the PSNR is the largest for patch size 40×40 with steps 10 and 20 pixels and patch size 80×80 with steps 20 and 40 pixels. Our experience with more example images shows that, from the point of view of both the visual effects and PSNR, the patch size 80×80 with step 20 and patch size 40×40 with step 10 usually achieve the best results. Despite the higher PSNR for patch sizes 40×40 and 80×80 with steps 20 and 40 respectively, than patch sizes 40×40 and 80×80 with steps 10 and 20 respectively, due to the more blurry its visual effects, the results of the last two patches are preferred. Since patch size 40×40 with step 10 is more time consuming than patch size 80×80 with step 20, we use the patch size 80×80 with step 20 in our following experiments for pose-invariant face recognition. Figure 4 shows three examples of generated virtual frontal views from CMU PIE face database with two angles, 22.5 and 45 degrees. In addition, we show four samples of generating the virtual frontal views for the pose yawing over 40 degrees which belong to NCKU CSIE face database (see Figure 5). According to our experimental results, we conclude that the LKNR method is an efficient and robust method for generating the virtual frontal view under different circumstances, i.e. different poses of individual persons.

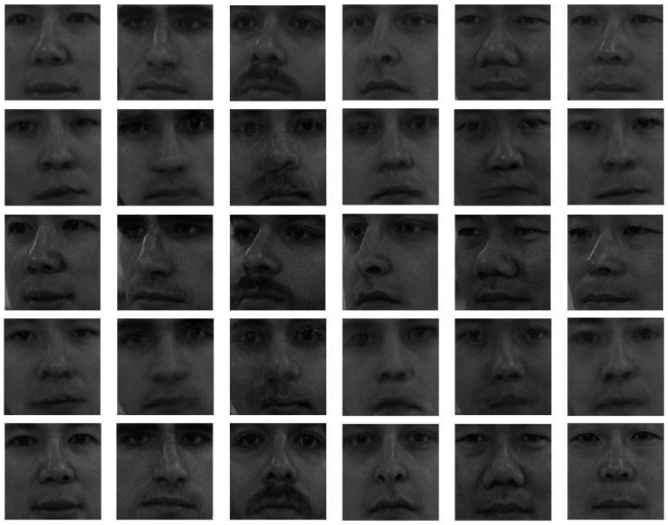


Figure 4. Examples of LKNR-based virtual frontal view generation from CMU PIE face database. From top to bottom, the upper row includes the original non-frontal faces with the pose yawing about 22.5 degree (P29), the second row is the generated virtual frontal views from P29, the third row includes the original non-frontal faces with the pose yawing about 45 degree (P11), the fourth row is the generated virtual frontal views from P11 and the last row shows the ground truth frontal images (P27)

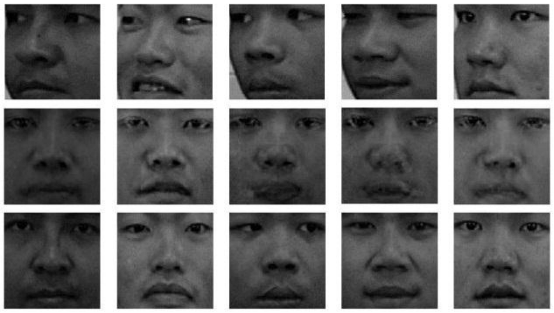


Figure 5. Examples of LKNR-based virtual frontal view generation from NCKU CSIE face database for the pose yawing about 40 degree. The upper row is the original non-frontal faces, the middle row is the virtual frontal views and the lower row is the original ground truth frontal views

4.2. Recognizing the Virtual Frontal Face

After generating the virtual frontal view from an image with non-frontal pose, it turns to use an efficient face recognition algorithm. Although, afterward the virtual frontal view is generated by LKNR we can employ any face recognition method, in this paper, we use the Fisher linear discriminant analysis (FLDA) [31], which is one of the most successful methods of classification, to validate the effectiveness of the proposed method. In general, FLDA is trying to find a linear transformation whereas the feature clusters are being separated well. This transformation is done through the scattering matrix analysis. So an input face image is transformed into a subspace where scattering the between-class is maximum and scattering the within-class is minimum by maximizing the Fisher separation criterion. When designing a FLDA classifier, one has to deal with the within-class scatter matrix carefully, because it may be singular. To avoid the singularity problem, PCA is first conducted to reduce the dimensionality to less than N-C, where N is the number of training examples, and C is the number of classes. The PCA transformed features are then fed into the final FLDA for classification. For this purpose, we use CMU PIE and FERET databases [32]–[33] as the testing and the training images, respectively. As our goal is performing the proposed face recognition algorithm on CMU PIE database, we use three subsets from the FERET pose database, i.e., “ba” (frontal), “be”(right rotation of 15 degrees) and “bf”(left rotation of 15 degrees), to form the training set in our experiments. Some example training images are shown in Figure 6. Due to the fact that the imaging conditions between the FERET and CMUPIE databases are different, the evaluation results on PIE database are expected to have a good generalizability. After PCA and FLDA models are obtained by using the FERET face images, the face recognition based on PCA+FLDA is performed on the 68 subjects in the CMU PIE dataset.

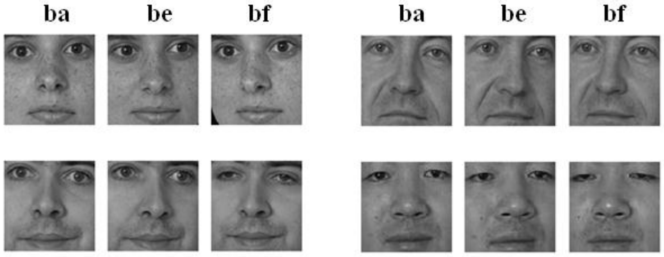


Figure 6. Example Images from the FERET Database for the Training of PCA+FLDA Method

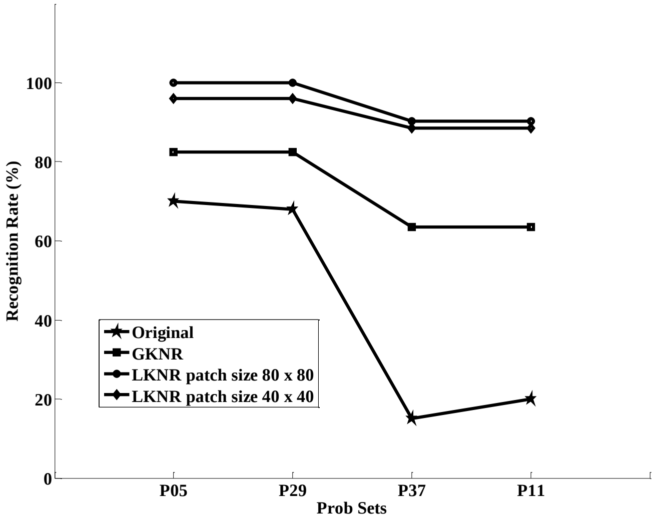


Figure 7. Comparing the Performance of Pose-invariant Face Recognition System PCA+FLDA without and with Kernel-based Nonlinear Methods

Table 2. The Face Recognition Rate of Our Proposed Algorithm and LLR and ELF Methods

Probe Sets	Methods				
	GKNR (%)	LKNR-step 20 Patch size 80×80 (%)	LKNR-step 10 Patch size 40×40 (%)	LLR-step 5 [2] (PCA+FLDA) (%)	ELF method [13] (3-P Normalization) (%)
P05	82.4	100	97.5	98.5	88
P37	63.5	90.3	89.4	82.4	74
P29	82.4	100	97.5	100	86
P11	63.5	90.3	89.4	89.7	76
Average	73.45	95.15	93.45	92.65	81

According to Figure 7, it is found that the recognition rate can be improved by using the virtual frontal view which was generated from the non-frontal face. Using the generated images via LKNR in the face recognition system the accuracy of system performance is increased. From Figure 7, since the quality of generated images via GKNR is lower than LKNR, therefore, its efficiency in face recognition is less, but however, GKNR has better performance than direct use of original non-frontal images in face recognition. In addition, since a small patch size may lead to more artifacts, especially the patch sizes 20×20 and 40×40, also given that these patches will consume more time, LKNR by using patch size 80×80 shows better results in different poses. Also, it is to be considered that the dense sampling can remove the blocking effects efficiently, therefore the step size of 20 pixels is the best choice where the patch size is 80×80 for LKNR method.

The ELF algorithm is the well known method for recognizing faces with different poses [13]. According to the results comparing the performance of LLR [2] and ELF methods that are written in Table 2, one can find that the proposed method with the step size equal 20 pixels and the patch size equal 80×80 outperforms the ELF on all probe sets. It should be noted that the results of LLR and ELF in Table 2 are directly cited from [2] and [13], respectively. Based on this, the average recognition rate of our method is 14.15% higher than the ELF method. In comparison with the implementation complexity, both LLR and LKNR need only the two eyes alignments, so they are simple than ELF. In addition, the main advantage of using LKNR in comparison with LLR is that it acts as a nonlinear method and it can handle nonlinear variations in images, therefore, the average recognition rate of our method is 2.5% higher than LLR and this is acceptable and not unexpected.

5. Conclusion

In this paper, we proposed a nonlinear method for generating the virtual frontal view from a non-frontal face image in order to improve the pose-invariant face recognition rate. As a linear method it may be inability when the face patterns are subject to large variations in viewpoints, illumination and expression. Indeed the linear assumption could dramatically decrease the computational consumption, but however, since the rotation of a human head is known as a nonlinear problem, the linear method eliminates lots of information. According to Cover’s theorem [26], nonlinearly separable patterns in an input space will become linearly separable with high probability if the input space is transformed nonlinearly into a high-dimensional feature space. So, we proposed a kernel-based nonlinear regression algorithm in order to achieve better results in face images with large variations, because the kernel functions can be used in any algorithm that solely depends on the dot product between two vectors, in other words, wherever a dot product is used it is replaced by a kernel function. Accordingly, with the help of kernel functions, we were overcome to the computational complexity of high dimensional nonlinear functions by embedding these functions in the stage of computing the reconstruction matrix from the non-frontal input face and non-frontal face database. After converting non-frontal face images into the virtual frontal view, we used the PCA+FLDA method for pose-invariant face recognition.

Acknowledgments

Portions of the research in this paper use the FERET database of facial images [32]–[33] collected under the FERET program, sponsored by the DOD Counterdrug Technology Development Program Office.

References

[1] Naseem, R. Togneri and M. Bennamoun, "Linear regression for face recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 11, **(2010)** November, pp. 2106-2112.

[2] X. Chai, S. Shan, X. Chen and W. Gao, "Locally linear regression for pose-invariant face recognition", IEEE Transactions on image processing, vol. 16, no. 7, **(2007)** July, pp. 1716-1725.

[3] R. Chellappa, C. L. Wilson and S. Sirohey, "Human and machine recognition of faces: Asurvey", in Proceeding of the IEEE, vol. 83, no. 5, **(1995)** May, pp. 705-740.

[4] W. Zhao, R. Chellappa, P. J. Phillips and A. Rosenfeld, "Face recognition: A literature surveys", ACM Computer Surveillance, vol. 35, no. 4, **(2003)**, pp. 399-458.

[5] P. Phillips, P. Grother, R. Micheals, D. Blackburn, E. Tabassi and M. Bone, "Face recognition vendor test 2002: Evaluation report", presented at the Face Recognition Vendor Test, **(2003)** March, NIST.

[6] M. Turk and A. Pentland, "Eigenfaces for recognition", Journal of Cognitive Neurology, vol. 3, no. 1, **(1991)** pp. 71-96.

[7] H. Murase and S. Nayar, "Learning and recognition of 3D objects from appearance", in Proceeding of the Qualitative Vision Workshop, New York, **(1993)** June, pp. 39-50.

[8] A. Pentland, B. Moghaddam and T. Starner, "View-based and modular eigenspace for face recognition", in Proceeding of the IEEE Conferance on Computer Vision and Pattern Recognition, Seattle, WA, **(1994)**, pp. 84-91.

[9] H. Murase and S. K. Nayar, "Visual learning and recognition of 3-D objects form appearance", International Journal of Computer Vision, vol. 14, no. 1, **(1995)**, pp. 5-24.

[10] S. McKenna, S. Gong and J. Collins, "Face tracking and pose representation", in Proceeding of the British Machine Vision Conference, Edinburgh, U.K, **(1996)**, pp. 755-764.

[11] Z. Zhou, J. HuangFu, H. Zhang and Z. Chen, "Neural network ensemble based view invariant face recognition", Journal of Computer Study Development, vol. 38, no. 9, **(2001)**, pp. 1061-1065.

[12] R. Gross, I. Matthews and S. Baker, "Eigen light-fields and face recognition across pose", in Proceeding of the 5th International Conference Automatic Face and Gesture Recognition, Washington, DC, **(2002)**, pp. 3-9.

[13] R. Gross, I. Mattews and S. Baker, "Appearance-based face recognition and light-fields", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 4, **(2004)** April, pp. 449-465.

[14] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces", in Proceeding of the Special Interest Group on Graphics and Interactive Techniques, New York, **(1999)**, pp. 187-194.

[15] V. Blanz and T. Vetter, "Face recognition based on fitting a 3-D morphable model", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 9, **(2003)** September, pp. 1063-1074.

[16] W. Zhao and R. Chellappa, "SFS based view synthesis for robust face recognition", in Proceeding of the 4th International Conference Automatic Face and Gesture Recognition, Grenoble, France, **(2000)**, pp. 285-292.

[17] G. Feng and P. C. Yuen, "Recognition of head-&-shoulder face image using virtual frontal view image", IEEE Transactions on Systems, Man and Cybernetics Part A: Systems and humans, vol. 30, no. 6, **(2000)** June, pp. 871-883.

[18] A. S. Georghiades, P. N. Belhumeur and D. J. Keiegman, "From few to many: Illumination cone models for face recognition under variable lighting and poses", IEEE Transactions on Pattern Analysis Machine Intelligence, vol. 23, no. 6, **(2001)** June, pp. 643-660.

[19] M. W. Lee and S. Ranganath, "Pose-invariant face recognition using a 3D deformable model", Pattern Recognition, vol. 36, no. 8, **(2003)**, pp. 1835-1846.

[20] D. Jiang, Y. Hu, S. Yan, L. Zhang, H. Zhang and W. Gao, "Efficient 3D reconstruction for face reconstruction", Pattern Recognition, vol. 38, no. 6, **(2005)**, pp. 787-798.

[21] T. Poggio and T. Vetter, "Recognition and structure from one 2-d model view: Observations on prototypes, object classes, and symmetries", Tech. Rep. A. I. Memo, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, no. 1347, **(1992)**.

[22] D. Beymer, "Face recognition under varying pose", Tech. Rep. A. I. Memo, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, no. 1461, **(1993)**.

[23] D. Beymer and T. Poggio, "Face recognition from one example view", in Proceeding of the 5th International Conference on Computer Vision, **(1995)**, pp. 500-507.

[24] T. Vetter and T. Poggio, "Linear object classes and image synthesis from a single example image", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, **(1997)** July, pp. 733-742.

[25] T. Vetter, "Synthesis of novel views from a single face image", International Journal of Computer Vision, vol. 28, no. 2, **(1998)**, pp. 103-116.

[26] T. M. Cover, "Geometrical and Statistical properties of systems of linear inequalities with applications in pattern recognition", IEEE Transactions on Electronic Computers, vol. EC-14, **(1965)**, pp. 326-334.

[27] B. Schölkopf, A. Smola and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem", Neural Computation, vol. 10, **(1999)**, pp. 1299-1319.

[28] B. Scholkopf and A. Smola, "Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond", Cambridge, MA: MIT, **(2002)**.

[29] T. Sim, S. Baker and M. Bsat, "The CMU pose, illumination, and expression (PIE) database", in Proceeding of the 5th International Conference Automatic Face and Gesture Recognition, Washington, DC, **(2002)**, pp. 46-51.

[30] Nat. Cheng Kung Univ., Tainan, Taiwan, "NCKU CSIE Robotics Lab face database", http://robotics.csie.ncku.edu.tw/Database.htm, **(2006)**.

[31] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, **(1997)** July, pp. 711-720.

[32] P. J. Phillips, H. Wechsler, J. Huang and P. Rauss, "The FERET database and evaluation procedure for face recognition algorithms", Image and Vision Computing Journal, vol. 16, no. 5, **(1998)**, pp. 295-306.

[33] P. J. Phillips, H. Moon, S. A. Rizvi and P. J. Rauss, "The FERET Evaluation Methodology for Face Recognition Algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, **(2000)**, pp. 1090-1104.

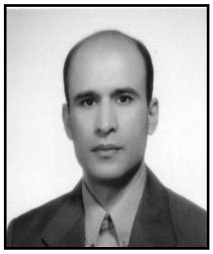
Authors

Yaser Arianpour received B.Sc. degree in electronic engineering from Islamic Azad University, Saveh branch, Iran, in 2008. He is currently pursuing the M.Sc. degree in electronic engineering from Islamic Azad University, south Tehran branch, Iran. His current research interests include signal and image processing and pattern recognition.

E-mail: st_y_arianpour@azad.ac.ir

Sedigheh Ghofrani received B.Sc. degree in electronic engineering from Tehran University, Iran, in 1991, the M.Sc. degree in communication from Islamic Azad University, south Tehran branch, Iran, in 1997 and Ph.D. in electronic from Iran University of science and technology, in 2004. She has been the assistant professor of Electrical Engineering Department at the Islamic Azad University, south Tehran branch since 2004. Her area of research includes image processing and signal processing.

E-mail: s_ghofrani@azad.ac.ir



Hamidreza Amindavar received B.Sc. degree in electrical engineering in 1985, the M.Sc. in electrical engineering in 1987, the M.Sc. degree in applied mathematics in 1991, and the Ph.D. degree in electrical engineering in 1991, all from University of Washington, Seattle, USA. He is currently a Professor of electrical engineering at Amirkabir University of Technology, Teheran, Iran. His research interests include signal and image processing, array processing, and multiuser detection. Dr. Amindavar is a member of Tau Beta Pi and Eta Kappa Nu.

E-mail: hamidami@aut.ac.ir