Optical Engineering

SPIEDigitalLibrary.org/oe

Feature space-based human face image representation and recognition

Yong Xu
Zizhu Fan
Qi Zhu

Feature space-based human face image representation and recognition

Yong Xu
Harbin Institute of Technology
Bio-computing Research Center
Shenzhen Graduate School and
Key Laboratory of Network Oriented Intelligent Computation
Shenzhen, China
E-mail: laterfall2@yahoo.com.cn

Zizhu Fan
Qi Zhu
Harbin Institute of Technology
Bio-computing Research Center
Shenzhen Graduate School
Shenzhen, China

Abstract. We propose a novel face recognition method that represents and classifies face images in the feature space. It first assumes that in the feature space the test sample can be well expressed by a linear combination of the training samples, and then it exploits the obtained linear combination to perform face recognition. We also present the foundation, rationale, and characteristics of, as well as the differences between, our method and conventional kernel methods. The analysis shows that our method is a representation-based kernel method and works in the feature space. This method might be able to outperform the representation-based methods that work in the original space. The experimental results show that our method partially possesses the properties of "sparseness" and is able to reduce greatly the effects of noise and occlusion in the test sample. [DOI: 10.1117/1.OE.51.1.017205]

Subject terms: image representation; classification; computer vision.

Paper 110778 received Jul. 5, 2011; revised manuscript received Oct. 18, 2011; accepted for publication Nov. 15, 2011; published online Feb. 7, 2012.

1 Introduction

Automated face recognition, which usually depends on a machine to recognize the identity of humans, has attracted much attention in recent years.[1–7] This technique can help the machine to identify people, interact with people, or provide better service for the host. If a machine not only can recognize the host but also identify his facial expression, it will be able to provide us with more help.[2,3] Indeed, automated face recognition enables machines to interact with people in a human way. Real-world applications of face recognition have much concern about the robustness[2] and high computational efficiency[4] of the algorithm. Nowadays high-performance hardware can allow the machine to satisfy the computational efficiency requirement of automated face recognition, whereas the robustness of previous face recognition algorithms was not good. Previous face recognition algorithms could not deal well with the noise and occlusion of the face. The noise is usually caused during the process of image capture and transmission. In a complex environment it is also possible that a portion of the face image is occluded by other objects. However, as shown in Sec. 4, our face recognition method is robust against occlusion and noise.

Automated face recognition[8–15] methods normally depend on a training process in which a set of training samples is used to obtain a solution and then the solution is used to classify test samples. However, a recently proposed pattern recognition method did not make use of any training procedure.[8,16] This method first decomposes a test sample as the sum of an error vector and a linear combination of the training samples. It then evaluates the contribution of the training samples of each class to represent the test sample, and classifies the test sample into the class that makes the greatest contribution to the representation. This method has been applied to solve problems in areas such as clustering, feature selection, face recognition, and signal processing.[16–23]

The method proposed in Refs. 8 and 16 suffers from the following problem: a test sample is represented by only a sparse linear combination of the training samples. In other words, when it represents the test sample with a linear combination of all the training samples, it assumes that a large portion of training samples have zero coefficient. For face recognition, the number of the training samples is usually much smaller than the dimensionality of the sample, and using the method in Refs. 8 and 16 to represent the test sample must cause representation error. Moreover, a large representation error might lead to incorrect classification of the test sample. The method also has a high time complexity.

In this paper, we propose a novel method that uses all the training samples in the feature space to represent and classify test samples. Because the kernel function is used, the proposed method is also a kernel-based method, referred to as a kernel and representation—based method (KRBM). We provide a detailed analysis of the proposed method, showing its foundation, rationale, characteristics, and time complexity. We also compare it with other kernel methods. Because the proposed method implicitly uses a nonlinear mapping to transform the samples into a new space, it is possible that the total new samples, generated by the implicit transform, are more representative of the sample space. The experimental results show that the proposed method not only is able to obtain a good classification result but also somewhat possesses the property of sparseness; that is, the majority of the coefficients of the linear combination have small absolute values. Sparseness has been shown to be helpful in achieving a good classification performance.[16] Though the method in Ref. 16 also achieves a sparse representation, it does so with a costly, deliberated, iterative solution scheme and a special constraint condition. Our method can be implemented easily at a low computational cost and without any constraint condition. A further advantage of our method is that it uses only

0091-3286/2012/$25.00 © 2012 SPIE

an n-dimensional vector to denote the original sample, where n is the number of the training samples. This means that our method might greatly reduce the dimension of the original image sample.

The paper is organized as follows: Sec. 2 describes the proposed method; Sec. 3 provides some analysis of our method; and Sec. 4 presents the experimental results. Section 5 offers our conclusions.

2 Methods

Our method first represents the test sample in the feature space through a linear combination of all the training samples and then uses the representation result to classify the test sample. Let $A_1 \ldots A_n$ denote n training samples in the original space. Let Y be a test sample in the original space. The feature space is derived from the original sample space by using a nonlinear mapping ϕ. If test sample $\phi(Y)$ in the feature space can be approximately described by a linear combination of all the training samples $\phi(A_1) \ldots \phi(A_n)$ in the feature space, then we have $\phi(Y) = \sum_{i=1}^{n} \beta_i \phi(A_i)$. Assuming that any sample in the feature space is a column vector, we can rewrite $\phi(Y) = \sum_{i=1}^{n} \beta_i \phi(A_i)$ into the following equation:

$$\phi(Y) = \Phi\Psi, \tag{1}$$

where $\Phi = [\phi(A_1) \ldots \phi(A_n)]$, $\Psi = (\beta_1 \ldots \beta_n)^T$. Since Φ might not be a square matrix and ϕ is unknown, we cannot directly solve Eq. (1). However, Eq. (1) has the following normal equation:

$$\Phi^T \phi(Y) = \Phi^T \Phi \Psi. \tag{2}$$

If we use the definition of kernel function $k(A_i, A_j) = \phi^T(A_i)\phi(A_j)$,[24,25] Eq. (2) can be transformed into

$$K_Y = K\Psi, \tag{3}$$

where

$$K_Y = \begin{pmatrix} k(A_1, Y) \\ \vdots \\ k(A_n, Y) \end{pmatrix}, \quad K = \begin{pmatrix} k(A_1, A_1) \ldots k(A_1, A_n) \\ \vdots \\ k(A_n, A_1) \ldots k(A_n, A_n) \end{pmatrix},$$

$$\Psi = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_n \end{pmatrix}.$$

If K is not singular, the solution of Eq. (3) can be solved by using $\Psi = K^{-1}K_Y$. If K is nearly singular, we can use $\Psi = (K + \mu I)^{-1}K_Y$ (μ is a positive constant and I is the identity matrix) to get the solution.

It is clear that $K_Y = (K_1 \ldots K_n)\Psi = \beta_1 K_1 + \ldots + \beta_n K_n$, where $K_i = [k(A_1, A_i) \quad k(A_2, A_i) \quad \ldots \quad k(A_n, A_i)]^T$. This shows that the issue of representing the test sample in the feature space has been transformed into a new issue of representing K_y by K_i, which denotes the ith column of matrix K. We refer to K_i as kernel vector of the ith training sample. It seems that training samples from different classes make different contributions to representing K_y. We evaluate the contribution of each class and classify the test sample as follows: first, we calculate the sum of the contribution of the training samples from each class. We assume that all the training samples from the kth class are $A_s \ldots A_t$. Thus the contribution to representing the test sample of the kth class is $g_k = \beta_s K_s + \ldots + \beta_t K_t$. The smaller the error $e_k = \|K_Y - g_k\|^2$ is, the greater the contribution of the kth class is. We identify the class that makes the greatest contribution to representing Y (i.e., the class that corresponds to the minimum error) and classify Y into the same class. This tends to classify the test sample into the class that is the most similar to this sample because a small e_k means that the linear combination of the training samples of the kth class is close to the test sample.

3 Analysis of Our Method and Comparison with Other Methods

In this section we will provide the rationale for our proposed method, describe its characteristics, and compare it with other methods.

3.1 *Rationale and Characteristics of Kernel and Representation–based Methods*

This subsection shows the rationale and characteristics of the KRBM. Supposing that the nonlinear mapping ϕ is known, we can solve Eq. (2) using $\Phi^T\phi(Y)$. It is easy to show that $\Phi^T\phi(Y) = [\phi^T(A_1)\phi(Y) \ldots \phi^T(A_n)\phi(Y)]^T$ and $(\Phi^T\Phi)_{ij} = \phi^T(A_i)\phi(A_j), i,j, = 1, 2, \ldots, n$. $\phi(Y)$ and $\phi(A_i)$ are the test sample and training sample, respectively, in the feature space. We note that $\phi^T(A_i)\phi(Y) = \|\phi(A_i)\| \cdot \|\phi(Y)\| \cos\theta_i$, where θ_i denotes the angle between vectors $\phi(A_i)$ and $\phi(Y)$. As a result, if all the samples in the feature space, including $\phi(A_i)$ and $\phi(Y)$, are unit vectors, then $[\phi^T(A_1)\phi(Y) \ldots \phi^T(A_n)\phi(Y)]^T$ will denote the cosine similarity between the test sample and each of training samples in the feature space. Provided that two arbitrary training samples are orthogonal in the feature space, both $\Phi^T\Phi$ and $(\Phi^T\Phi)^{-1}$ will be the identity matrices. We then can obtain $\Psi = [\phi^T(A_1)\phi(Y) \ldots \phi^T(A_n)\phi(Y)]^T = [\cos\theta_1 \ldots \cos\theta_n]^T$. This shows that, under these assumptions, the solution vector of our method consists of n components that represent the cosine similarity between training samples and the test sample in the feature space. As shown in Sec. 2, the issue of using a linear combination of all the training samples to represent the test sample can be formulated by $K_Y = \beta_1 K_1 + \ldots + \beta_n K_n$. The experimental result shows that if in the feature space the ith training sample has a large similarity with the test sample, β_i usually is large. As a result, the contribution of the ith training sample, i.e., $\beta_i K_i$, also is large. If the training samples from one class are similar to the test sample, we can say that the class is similar to the test sample. Based on the classification rule shown earlier, the KRBM tends to classify the test sample into the class that is the most similar to it. This partially demonstrates the rationale of the KRBM.

Our method has two characteristics. First, it produces one linear system and exploits the solution of the system to classify the test sample. Second, it solves the linear system at a low cost of time complexity. This can be shown as follows: supposing that each class has m training samples and the number of all the training samples is $n = Lm$, where L is the number of the classes, we know that our method should

solve only one linear system in the form of Eq. (3) with a time complexity of $O(n^3 + n^2)$.

We analyze the computational cost of the sparse representation method proposed in Ref. 16 as follows. Because this method uses an iterative algorithm to solve a linear programming problem, it will have a high time complexity. Even if we do not take the iteration steps into account and calculate only the complexity of solving one linear system, the sparse representation method still has a time complexity of $O(n^2M + n^3 + nM)$, where M is the dimension of the sample vector in the original space. Consequently, it is clear that the sparse representation method will need a much higher time complexity to solve the linear system than the KRBM.

As will be shown in Sec. 4, our method also partially possesses the property of sparseness, as does the sparse representation method proposed in Refs. 8 and 16. That is, the majority of the components of the solution vector of our method have small absolute values. The literature provides evidence that sparseness is helpful in achieving a good classification performance. A particular advantage of the KRBM is that, unlike that in Ref. 16, it exhibits sparseness yet requires no extra cost.

3.2 *Comparison of Our Method with Other Methods*

In this subsection we first will compare the KRBM with the method proposed in Ref. 8 from the viewpoint of sample representation. We then will provide a comparison of our proposed method and conventional kernel methods. We note that the method proposed in Ref. 8 attempted to denote the test sample by a sparse linear combination of the training samples in the original space. In other words, it tried to make the majority of the coefficients of the linear combination that express the test sample equal or close to zero. Almost all the sparse representation methods work in the original space.[26–30] Theoretically, under the condition that the training samples in the original space can construct the basis of this space, the method proposed in Ref. 8 would be able to represent the test sample without error. However, in real-world applications, this condition is usually not satisfied and the representation almost always causes an error. A large error means that the test sample is poorly represented. It also seems that the fewer training samples used, the greater the error. If the error of sample representation is too large, the test sample might be not able to be classified correctly. To our knowledge, in the original space, the face recognition problem is usually a high-dimensional problem in which the number of training samples is smaller than the dimensionality of samples, and training samples are often correlated, which implies that there usually exists a large representation error in the test sample. Nevertheless, in the feature space, the training samples might become truly or approximately uncorrelated because of the use of nonlinear mapping. As a result, in this case the training samples in the feature space can provide more information than those in the original space, and our method seems to be able to more accurately represent the test sample than the method presented in Ref. 8. We note that sparse kernel density estimation has been proposed.[31]

Here we compare our method with conventional kernel methods[24,25] such as kernel principal component analysis[32]

Fig. 1 Some face images of one subject in the AR database.

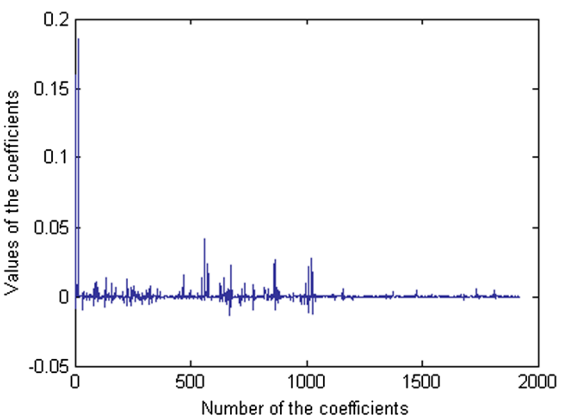

Fig. 2 The coefficients obtained using our method on the first test sample from the AR face database.

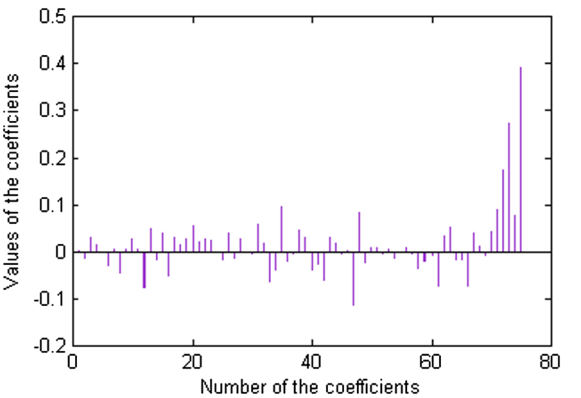

Fig. 3 The coefficients obtained using our method on the last test sample from the Yale face database.

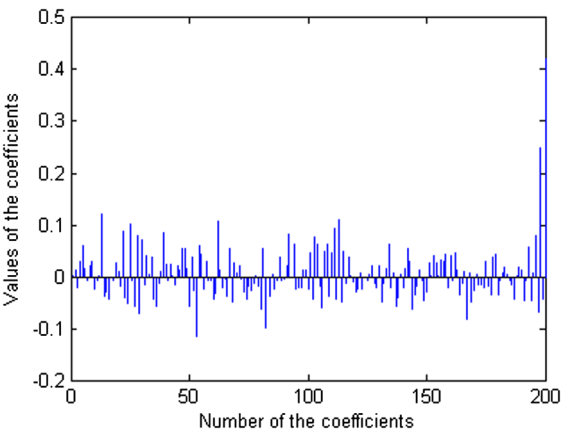

Fig. 4 The coefficients obtained using our method on the last test sample from the ORL face database.

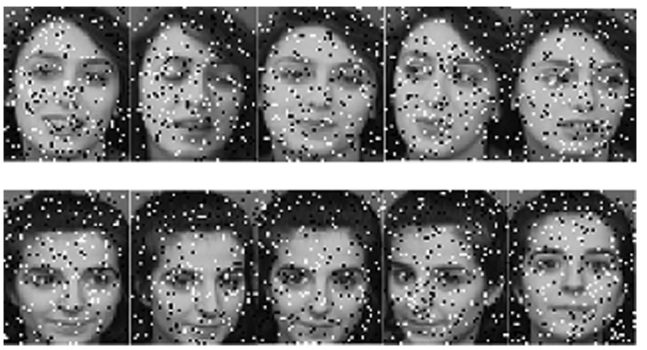

Fig. 5 Ten test images with random salt and pepper noise. Ten percent of the pixels were corrupted by the noise.

and kernel Fisher discriminant analysis (KFDA).[33] These kernel methods are typical examples of kernel-based transform methods and the transform result of the test sample in the feature space is its projection onto the eigenvector or discriminant vector in the feature space. As a result, each component of the transform result will be a linear combination of n kernel functions, where n is the number of the training samples. It is clear that, when representing test samples of conventional kernel methods, one must consider that different test samples can be represented by a linear combination of different kernel functions, and these different test samples share the same linear combination coefficients.

Our method also can be viewed as a transform-based kernel method. It first implicitly transforms the test sample into a new space and then tries to express the test sample in the new space. As shown earlier, using the kernel function, we can convert the issue of expressing the test sample into an issue defined as in Eq. (3). Once the Ψ in Eq. (3) is obtained, the coefficients of the linear combination are determined. As a result, we also can consider $K_Y = \beta_1 K_1 + \ldots + \beta_n K_n$ as the transform result of test sample Y. We note that both β_i and K_i $(i = 1, 2, \ldots, n)$ vary with Y. In other words, when our method uses a linear combination of the training samples in the feature space to represent the test sample, both the kernel function and coefficient vary with the test sample. This makes our method very different from the conventional kernel methods. Moreover, because our method constructs a special linear combination to represent the test sample, it is better able to represent the test sample than the conventional kernel methods.

4 Experimental Results

We used the AR,[34,35] Olivetti Research Laboratories (ORL),[36] and Yale[37] face database to test our method. Figure 1 shows some face images from the AR database. We adopted the Gaussian kernel function $k(x_i, x_j) = \exp[-\|x_i - x_j\|^2/(2\sigma)]$, where σ is the parameter of the kernel function. We partitioned each face database into a training set and test set. We took the first five face images per class of the ORL and Yale databases as training samples and treated the remainder as the test samples. We took the first 16 face images per class and the remainder of the AR database as training samples and the test samples, respectively. We first resized each face image into half of the original size by using the down-sampling method provided in Ref. 32. Because our method is directly applicable for only a one-dimensional vector, we converted the face image sample into a one-dimensional vector in advance. Because these one-dimensional vectors are high dimensional, we reduced the dimensionality by exploiting principal component analysis (PCA) to transform these vectors into $n - 1$ dimensional vectors, where n is the number of all

Table 1 Classification error rates of the proposed method on original test samples from the face databases.

Database	Training Samples Per Class (n)	Our Method	PCA	LDA	KFDA*
ORL	5	0.08 (1.0e7)	0.090	0.150	0.100(1.0e7)
Yale	5	0.044 (1.0e7)	0.067	0.044	0.122(1.0e7)
AR	16	0.254 (1.0e6)	0.303	0.725	0.396 (1.0e7)

The number in parentheses denotes the value of the parameter of the kernel function.
*As shown in Ref. 33.

Table 2 Classification error rates of the proposed method on face databases in which the test sample was corrupted by "salt and pepper" noise.

Database	Training Samples Per Class (n)	Our Method	PCA	LDA	KFDA*
ORL	5	0.095 (1.0e7)	0.110	0.195	0.340 (1.0e7)
Yale	5	0.044 (1.0e7)	0.089	0.067	0.189 (1.0e7)
AR	16	0.268 (1.0e6)	0.308	0.979	0.459 (1.0e7)

The number in parentheses denotes the value of the parameter of the kernel function.
*As shown in Ref. 33.

Fig. 6 Some occluded test samples of one subject from the AR database. The first image is the baboon image, a part of which is used as the mask that occludes the face image. The other images are the occluded test samples.

Fig. 7 Some occluded test samples of one subject from the Yale database. The first image is the peppers image, a part of which is used as the mask that occludes the face image. The other images are the occluded test samples.

the training samples, and then carried out our method on the obtained vectors.

Figures 2, 3, and 4, respectively, show the coefficients obtained using our method on one test sample from the AR, Yale, and ORL face databases. It can be seen that our method somewhat possesses the properties of sparseness; that is, the majority of the coefficients of the linear combination expressing the test sample are equal or close to zero. Our method is different from the method proposed in Ref. 38 as follows: first, our method represents and classifies the test sample in the feature space, whereas the method in Ref. 38 does the same thing in the original space. Second, the method in Ref. 38 uses the elaborately devised two-phase representation and classification scheme to achieve sparseness, whereas our method possesses the property of sparseness without any constraint imposed.

We then performed an experiment to show the effect of noise on our method. We added salt and pepper noise to each test image. Figure 5 shows some examples of the corrupted test images. Tables 1 and 2 show classification error rates of our method—PCA, linear discriminant analysis (LDA) and KFDA as in Ref. 35—on the original and corrupted test samples, respectively. When implementing PCA and LDA, we first used $n-1$ and $L-1$ transform axes, respectively (where n is the total number of training samples and L is the number of classes), to transform the samples into the new space; we then classified the test samples using the nearest neighbor classifier. Our method is not only able to classify more accurately but also is much more robust than the other methods. However, the other methods obtain much higher classification error rates for corrupted test samples than for original test samples. Moreover, LDA failed almost completely in classifying the corrupted face images.

We also performed experiments on partially occluded test samples to show the face recognition performance of our method under the real-world condition of partial occlusion of the face by other objects. The ratio of the occluded area to the whole image was 20% and the region to be occluded was set randomly. Figures 6–8 show some occluded test samples from different face databases. Table 3 shows the classification error rates of our method. This experiment again shows that our proposed method is more robust than the other methods. In summary, it is safe to say that the proposed method usually outperforms PCA, LDA, and KFDA.

We also conducted experiments to compare our method with the PCA plus LDA method and sparse representation method proposed in Ref. 39, which is indeed equivalent to the method presented in Ref. 8. For the ORL and Yale face databases, we performed the fivefold cross-validation experiments. Because in the Yale face database each subject provided 11 face images, we discarded the last face image of each subject and used only the remaining 10 images. For the AR database, we used the same training and test samples presented in the above context and did not exploit the cross-validation scheme. We used Table 4 to show the mean of the classification error rates of the proposed method, the PCA plus LDA method, and the sparse representation method proposed in Ref. 39, and we can see that our method outperforms the PCA plus LDA method and the sparse representation method.

Fig. 8 Some occluded test samples of two subjects from the ORL databases. The first image is the Lena image, a part of which is used as the mask that occludes the face image. The other images are the occluded test samples.

Table 3 Classification error rates of the proposed method on face databases in which the test sample was partially occluded*

Database	Training Samples Per Class (n)	Our Method	PCA	LDA	KFDA†
ORL	5	0.1550	0.160	0.255	0.200 (1.0e7)
Yale	5	0.0556	0.089	0.056	0.144 (1.0e7)
AR	16	0.2817	0.310	0.935	0.787 (1.0e7)

*The ratio of the occluded area to the whole image is 20%.
†As shown in Ref. 33.

Table 4 Performance comparison (classification error rates) of the proposed method, the PCA plus LDA method, and the sparse representation method.*

Database	Our Method	PCA Plus LDA	Sparse Representation Method
ORL	0.013	0.031	0.015
Yale	0.00	0.035	0.027
AR	0.254	0.313	0.276

*As proposed in Ref. 39. For the ORL and Yale face databases, we performed the fivefold cross-validation experiments. For the AR database, we used the same training and test samples presented in the above context.

5 Conclusions

The method proposed in this paper can be viewed as one that first decomposes the test sample into the weighted sum of all the training samples and then classifies the test sample into the class whose training samples make the greatest contribution to the weighted sum. The analysis partially explicates the foundation, rationale, and characteristics of the proposed method, which performs well in face recognition, producing a low classification error rate. The recognition performance of the method is also robust to noise and the occlusion of face images.

Our analysis clearly reveals the similarities and difference between the proposed method and previous kernel methods. A final point of interest about this research is that although it was not devised to satisfy the requirement of sparseness, it does possess this property (i.e., the majority of the elements of the solution vector are equal or close to zero). Compared with the sparse representation method proposed in Ref. 8, our method is simple, can be solved easily, and has a low computational cost.

Acknowledgments

This article is partly supported by the Program for New Century Excellent Talents in University (grant nos. NCET-08-0156 and NCET-08-0155); the NSFC under grant nos. 61071179, 60803090, 60902099, and 61001037; the Fundamental Research Funds for the Central Universities (HIT.NSRIF. 2009130); as well as Jiangxi Provincial Natural Science Foundation of China (grant no. 2010GQS0027).

References

1. G. C. Littlewort et al., "Towards social robots: automatic evaluation of human-robot interaction by face detection and expression classification," in *Advances in Neural Information Processing Systems*, S. Thrun, L. Saul, and B. Schoelkopf, Eds., vol. **16**, pp. 1563–1570, MIT Press, Cambridge, Mass (2004).
2. H. K. Ekenel et al., "Face recognition for smart interactions," in *Proc. of the 2007 IEEE International Conference on Multimedia and Expo*, IEEE Computer Society, Beijing, China, pp. 1007–1010 (2007).
3. H. K. Ekenel, M. Fischer, and R. Stiefelhagen, "Face recognition in smart rooms," in the *MLMI 2007 Proc. of the 4th International Conference on Machine Learning for Multimodal Interaction*, A. Popescu-Belis, H. Bourlard, and S. Renals, Eds., Springer-Verlag Berlin, Heidelberg, Germany, pp. 120–131 (2007).
4. C. Cruz, L. E. Sucar, and E. F. Morales, "Real-time face recognition for human-robot interaction," in *Proc. of the 8th. IEEE International Conference on Automatic Face and Gesture Recognition*, IEEE Computer Society, Amsterdam, The Netherlands, pp. 1–6 (2008).
5. K. Fukui and O. Yamaguchi, "Face recognition using multiviewpoint patterns for robot vision," *Robot. Res.* 192–201 (2003).
6. E. Erzin et al., "Multimodal person recognition for human-vehicle interaction," *IEEE Multimedia.* **13**(2), 18–31 (2006).
7. A. Couture-Beil, R. Vaughan, and G. Mori, "Selecting and commanding individual robots in a vision-based multi-robot system," presented at the *Seventh Canadian Conference on Computer and Robot Vision*, Ottawa, Ontario, Canada, 31 May to 2 June 2010.
8. J. Wright et al., "Sparse representation for computer vision and pattern recognition," in *Proc. IEEE*, 1031–1044 (2010).
9. D. Zhang et al.., "Advanced pattern recognition technologies with applications to biometrics," Hershey, Pa.: Medical Information Science Reference (2009).
10. Z. Jin et al., "Face recognition based on the uncorrelated discriminant transformation," *Pattern Recognit.* **34**(7), 1405–1416 (2001).
11. P. Belhumeur, J. Hespanda, and D. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(7), 711720 (1997).
12. Y. Xu et al., "An approach for directly extracting features from matrix data and its application in face recognition," *Neurocomputing* **71**(10–12), 1857–1865 (2008).
13. Y. Xu et al., "An efficient renovation on kernel Fisher discriminant analysis and face recognition experiments," *Pattern Recognit.* **37**, 2091–2094 (2004).
14. W. Zuo et al., "BDPCA plus LDA: a novel fast feature extraction technique for face recognition," *IEEE Trans. Syst. Man Cybern. B Cybern.* **36**(4), 946–953 (2006).
15. Y. Xu, L. Yao, and D. Zhang, "Improving the interest operator for face recognition," *Expert System With Applications* **36**(6), 9719–9728 (2009).
16. J. Wright et al., "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(2), 210–227 (2009).
17. C. X. Ren and D. Q. Dai, "Sparse representation by adding noisy duplicates for enhanced face recognition: an elastic net regularization approach," presented at the *Chinese Conference on Pattern Recognition*, 5–6 November 2009, Nanjing, China.
18. E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proc. IEEE Int. Conf. on Comput. Vis. and Patt. Recog.*, IEEE Computer Society, Miami, FL, USA (2009).
19. Y. Shi et al., "Sparse discriminant analysis for breast cancer biomarker identification and classification," *Prog. Nat. Sci.* **19**(11), 1635–1641 (2009).
20. E. Candes and T. Tao, "Near-optimal signal recovery from random projections: universal encoding strategies?," *IEEE Trans Inform. Theory* **52**(12), 5406–5425 (2006).
21. E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure Appl. Math.* **59**(8), 1207–1223 (2006).
22. Y. Shi et al., "Gene selection and visualization based on sparse maximal margin features," Presented at the *Chinese Conference on Pattern Recognition*, Nanjing, China, 5–6 (November 2009).
23. D. Donoho, "For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution," *Comm. Pure Appl. Math.* **59**(6), 797–829 (2006).
24. Y. Xu, D. Zhang, Z. Jin, M. Li, and J.-Y. Yang, "A fast kernel-based nonlinear discriminant analysis for multi-class problems," *Pattern Recognit.* **39**(6), 1026–1033 (2006).
25. K.-R. Muller et al., "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.* **12**(1), 181–201 (2001).
26. S. Yang, Z. Liu, and L. Jiao, "Multitask dictionary learning and sparse representation based single-image super-resolution reconstruction," *Neurocomputing* (2011).
27. M. Fan et al., "Sparse regularization for semi-supervised classification," *Pattern Recognit.* **44**(8), 1777–1784 (2011).
28. M. Yang et al., "Robust sparse coding for face recognition," *Proc. of the IEEE Conference on Computer Vision and Pattern Recognit* **2011**, 625–632 (2011).
29. X.-L. Wang et al., "Salt-and-pepper noise removal based on image sparse representation," *Opt. Eng.* **50**, 097007 (2011).
30. G. Chen, G. Dudekm, and L. A. Torres-Méndez, "Scene reconstruction with sparse range data and intensity information," *Opt. Eng.* **50**, 097002 (2011).
31. S. Chen, X. Hong, and C. J. Harris, "Regression-based D-optimality experimental design for sparse kernel density estimation," *Neurocomputing* **73**(4–6), 727–739 (2010).
32. Y. Xu et al., "A method for speeding up feature extraction based on KPCA," *Neurocomputing* **70**(4–6), 1056–1061 (2007).
33. S. Mika et al., "Fisher discriminant analysis with kernels," in *Proc. IEEE Signal Process. Soc. Workshop, Neural Networks for Signal Processing IX*, IEEE Signal Process Society, Madison, WI , USA, pp. 41–48 (1999).
34. http://cobweb.ecn.purdue.edu/~aleix/aleix_face_DB.html.
35. J. Yang , "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Trans. Pattern A Anal. Mech. Intell.* **26**(1), 131–137 (2004).

36. AT&T Laboratories, "The database of faces," 2002, http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html (29 December 2011).
37. "The Yale face database," http://cvc.yale.edu/projects/yalefaces/yalefaces.html (29 December 2011).
38. Y. Xu et al., "A two-phase test sample sparse representation method for use with face recognition," *IEEE Trans. Circuits Syst. Video Tech.* **21**(9), 1255–1262 (2011).
39. E. Candes and J. Romberg, "l1-Magic: recovery of sparse signals via convex programming," *Advances* 1–19 (2005).

Yong Xu received his BS degree in 1994 and his MS degree in 1997. He received a PhD degree in pattern recognition and intelligence systems at Nanjing University of Science and Technology, Nanjing, China, in 2005. Now he works at Shenzhen Graduate School, Harbin Institute of Technology. His current interests include feature extraction, biometrics, face recognition, machine learning, image processing, and video analysis.

Zizhu Fan received an MS degree in computer science from Hefei University of Technology, Hefei, China, in 2003. Now he is working on his PhD degree in computer science and technology at Shenzhen Graduate School, Harbin Institute of Technology. His current interests include pattern recognition and machine learning. He has had more than 10 articles published.

Qi Zhu obtained his Master's degree in 2009 from the Harbin Institute of Technology. He is working toward his PhD degree at Shenzhen Graduate School, Harbin Institute of Technology. His current interests include feature extraction and face recognition.