

Research Article

A Fast Iterative Pursuit Algorithm in Robust Face Recognition Based on Sparse Representation

Zhao Jian,¹ Huang Luxi,¹ Jia Jian,² and Xie Yu¹

¹ School of Information Science and Technology, Northwest University, Xi'an 710069, China

² Department of Mathematics, Northwest University, Xi'an 710069, China

Correspondence should be addressed to Zhao Jian; zjctec@nwu.edu.cn

Received 3 October 2013; Revised 19 December 2013; Accepted 28 December 2013; Published 13 February 2014

Academic Editor: Chung-Hao Chen

Copyright © 2014 Zhao Jian et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A relatively fast pursuit algorithm in face recognition is proposed, compared to existing pursuit algorithms. More stopping rules have been put forward to solve the problem of slow response of OMP, which can fully develop the superiority of pursuit algorithm—avoiding to process useless information in the training dictionary. For the test samples that are affected by partial occlusion, corruption, and facial disguise, recognition rates of most algorithms fall rapidly. The robust version of this algorithm can identify these samples automatically and process them accordingly. The recognition rates on ORL database, Yale database, and FERET database are 95.5%, 93.87%, and 92.29%, respectively. The recognition performance under various levels of occlusion and corruption is also experimentally proved to be significantly enhanced.

1. Introduction

Face recognition is one of the most active and challenging subject in computer vision and artificial intelligence, which has a wide range of applications such as personnel sign system, image search engine, and convicts detecting system. It has been experimentally proved that various sparse representation methods perform well in face recognition [1]. Given sufficient face images of k object members and a test image y , which belongs to one of the object classes, the problem of face recognition can be transformed into a classification issue [1, 2]. The basic idea of this kind of algorithms is to find a sparsest solution to represent y for classification [3, 4].

But current algorithms based on sparse representation also have drawbacks. On one hand, convergence speeds of these methods are slow to some extent. On the other hand, when the test image is under large percent of random corruption or contiguous occlusion, the sparse solution becomes denser so that it is hard for the system to find the right class y belongs to [5–7].

In this paper, a fast pursuit algorithm is proposed to solve the problem mentioned above. A common problem of pursuit algorithms is that the computational speed is quite slow. Aiming at face recognition, we improved the greedy algorithm,

making it much faster than kindred ones. By focusing on crucial information to classification issue—sparsity of errors, this algorithm enhances the robustness in the problem of face misalignment and large percent of occlusion. The basic frame of this paper is as follows: to begin with, we briefly review existing techniques for face recognition, including its advantages and brittleness to occlusion. Then, an improved algorithm is proposed and its feasibility and effectiveness will be demonstrated. Finally, we present experiments on ORL, Yale, and FERET face databases, as well as on a face database collected by ourselves, to verify the modified algorithm.

2. A Review of Pursuit Algorithms to Solve Face Recognition Issue

Pursuit algorithms are for solving the problem (p_0)

$$(p_0) : \min_x \|x\|_0 \quad \text{subject to } y = Ax, \quad (1)$$

where y represents the test sample, A is the training dictionary of the k known classes, and x is the sparse solution [8–10]. The core idea of these greedy algorithms is to update support and provisional solution iteratively in order to reduce the residual to a minimum [11]. Suppose the matrix A is

composed of n training face images of the k subjects, where an image is represented by a column of $A(m * n)$,

$$A = [A_1, A_2, \dots, A_k] = [v_{1,1}, v_{1,2}, \dots, v_{k,n_k}], \quad (2)$$

where A_i represents the i th class of A and $v_{i,j}$ represents j th element in A_i . And the given test sample y , well aligned, can be seen as one of the linear combination of these n -dimensional column vectors:

$$y = \sum_{i=1}^n a_{1,i} v_{1,1} + a_{2,i} v_{1,2} + \dots + a_{k,i} v_{k,n_k}, \quad (3)$$

where $a_{i,j}$ is the corresponding coefficient of v_{i,n_i} . The task of these algorithms is to find the sparsest solution so that class of the largest coefficient can be considered of the same class as y [12].

Now, let us briefly review the procedure of OMP and MP: firstly, find the i th column which can minimize $\varepsilon(i) = \|a_i v_i - y\|_2$, add this column to support, and compute a_i to make the residual minimized. Then, continue to update support and provisional solution iteratively until the residual is less than a threshold we set before. As to MP, which is similar to OMP, the apparent difference is that the coefficients of S^{k-1} original entries remain unchanged, rather than solving a least square for reevaluating all the update support stages.

The greedy strategy expands the support set, initially empty, by one additional column. So the enumeration takes n^{k_0} steps if the optimization problem is known to have k_0 nonzero members, which seems quite slow in many situations. But to the problem of face recognition, since the test sample y belongs to one class in A , in another word, correlated with only a few columns in A , k_0 is a controllable whose value is small. Many iterative-shrinkage algorithms, which “shrink” entry of every column in A to update the optimal solution iteratively in order to make the solution become “sparser,” additionally process $(n - 1)$ classes of useless information every stage. Compared to them, pursuit algorithms can find the most possible class of test image y by the foremost iterations. Just because of this extraordinary nature, we hold that the pursuit method is best for classification via sparse representation.

3. A Fast and Robust Pursuit Algorithm

One of the apparent drawbacks of pursuit algorithm is that the computing speed is slow in many situations, since it must continuously update the support and optimal solution until the result is accurate enough. In this section, we propose an improved pursuit algorithm for face recognition, which mainly involved two aspects: (1) general stopping rules to make the greedy algorithm faster in face recognition; (2) stopping rules for well-aligned and noiseless input samples.

Most state-of-the-art algorithms dealing with sparse representation try to find linear combination of matrix A , which approximates to input vector y . Goal of many kinds of iterative shrinkage algorithms is to “shrink” the sparse solution [13], in other words, to make the solution become “sparser” [14]; on the contrary, pursuit strategy is to make

the solution from sparse to dense. It is known that the basic idea of pursuit algorithm is to add columns to the support and update provisional solution until the residual between the proposed solution and the input vector y is small enough. We can easily imagine that it is hard to represent y by the linear combination of columns in A accurately when the test sample y is occluded or corrupted, as the distance between y and the corresponding column in A increases, and y may be relevant with more columns in A [15].

3.1. General Stopping Rules. Since test image y is related with just a few columns in matrix A , which belong to the same class, only coefficients of one class are valid to the recognition result. So it is unnecessary and inefficient to recover y precisely, which will unavoidable involve many irrelevant classes. Rather than recovering y accurately, our goal is to find the right class rapidly in face recognition; therefore more stopping rules are needed to make the algorithm faster.

3.1.1. Maximum Iterating Times. Let us consider the best situation first. If the test image y is the same as one column in matrix A , indicating that y and a_i are parallel, only one iteration which needs n flops to find that the maximum $(a_i^T y)^2$ is required to identify the class. When the test image is under random corruption or varying level of contiguous occlusion, the errors between y and every column of A become larger. Then y may be correlated with more classes in A . But the final recognition result depends on the largest coefficients of sparse solution, which suggest the most possible class of y . Hence even under the worst condition, n iterating times are enough because the iteration can be seen as an ergodic one to a classification problem where only one class is valid.

3.1.2. Results Resemblance of Two Successive Iterations. As what we have discussed above, our final identification result depends on the largest entries of the sparse solution; therefore we can neglect the details of the solution [6]. The minor change of the solution x may affect the accuracy of representation, but this would not influence the classification result. So if $\|x_k - x_{k-1}\|_2$ is smaller than some predetermined threshold, the iteration process can be stopped.

3.2. Stopping Rules and Processing Aiming at Different Image Samples. Under most conditions, we would not hope that the iteration times reach its upper limit: on one hand, it may take quite a long time; on the other, since the solution becomes denser when iteration times increase, it is harder to identify the right class. And the basic stopping rule of OMP, $\|r\|_2 < \varepsilon_0$, where $r = y - Ax$ and ε_0 is the error threshold, is often hard to obtain as test image may contain various noises or be disguised. Hence, more stopping rules should be devised to reduce iteration times. Now, we discuss this issue in two contexts—well-aligned, noiseless input samples and corrupted or occluded images.

3.2.1. Maximum Coefficient of the Sparse Solution. Suppose the input image contains little noise and is well-aligned, it is easy for the system to classify this sample. What we should do

is to raise the identification speed. The coefficients of sparse solution

$$x_i = \frac{a_i^T r}{\|a_i\|_2}, \quad \text{where } r = y - Ax^{k-1}, \quad (4)$$

x_i reflects the degree of resemblance between test sample y and one column in A — a_i . If the maximum coefficient is large enough, in other words, approximating to 1, we can consider that y belongs to the class of x_i , since y and x_i have already been sufficiently similar and we do not need to represent the minor errors in the linear combination of other columns as the small coefficients are senseless to classification and this process will surely increase the iteration times.

3.2.2. Sparse Level of the Sparse Solution. To corrupted or disguised test samples, however, large coefficients of the sparse solution are hard to reach owing to relatively larger range of noise. But the special characteristic of recognition or identification ensures that the recognition system can find the class of y before the representation error reaches the upper bound we have set. One of the key points is the sparse level of the sparse solution. We define the sparse level as the ratio of two largest coefficients which are in different classes,

$$k = \frac{a_i}{a_j}, \quad (5)$$

where a_i is the largest coefficient, which belongs to class i and a_j is the second largest coefficient whose class is different from class i .

3.2.3. The Noise Level. We define $\mu(A)$ as the maximal inner product between columns in A and test sample y ,

$$\mu(A) = \max [a_i^T \cdot y] \quad (6)$$

which reflects the maximum correlation between y and face images in training dictionary A . If $\mu(A)$ is smaller than some predefined threshold, it means that similarity between y and each column of A is less than that threshold, which indicates that the input sample is not a valid one or with much additional noise. Suppose the test sample is a valid one, it is inefficient and laborious for the system to process y directly since our goal is to try to find the one resemble y in the dictionary, but y itself is not precise enough. Both the random noise and the part of occlusion can be regarded as noise in the test image accordantly. So $\mu(A)$ implies the noise level of the input image.

3.2.4. Stopping Rules for Well-Aligned and Input Samples with Little Noise. The entries of sparse solution x reflect how much their respective contribution to the solution in respect that x includes all coefficients of the combination of columns in A . Matching pursuit algorithm update support and solution based on minimizing the errors

$$\varepsilon(i) = \min \|a_i x_i - r^{k-1}\|_2^2, \quad (7)$$

where $r = y - Ax^{k-1}$ and $x_i = a_i^T r^{k-1} / \|a_i\|_2^2$. For input samples with noise or misaligned, obviously, the coefficients of x decrease. So the largest entry threshold cannot be reached. But if the corrupted percent of the input image y is not too large, the corresponding level of y and a_i , which is in the same class as y , is relatively large compared to the others. This means that the solution is still quite sparse after a few iterations and the class of largest entry can be considered as the identification result. Using the concept of sparse level which we have defined before, the iteration processing can be stopped when a preset sparse level threshold reaches after some iteration times.

Proof. It can be proved that for a system of linear equations $Ax = y$, if a solution x exists obeying

$$\|x\|_0 \leq \frac{1}{2} \left(1 + \frac{1}{\mu(A)} \right). \quad (8)$$

OMP run with threshold parameter $\varepsilon_0 = 0$, is guaranteed to find it exactly. This theorem is only valid when test sample y can be represented by the linear combination of A exactly,

$$y = Ax = \sum x_i a_i. \quad (9)$$

Suppose that, $\mu(A)$ is equal or greater than some value close to 1, and after some iterations, the sparse level of the solution is still greater than p , and we only reserve the $(1 + 1/\mu(A))/2$ greater entries; hence

$$y = \left\{ 0, 0, \dots, x_i, 0, \dots, \frac{x_i}{p}, \dots \right\}. \quad (10)$$

It can be proved that if a sparse vector satisfies the sparsity constraint $\|x\|_0 < (1 + 1/\mu(A))/2$ and gives a representation of y to within error tolerance ε , every solution x must obey

$$\|x^\varepsilon - x\|_2^2 \leq \frac{4\varepsilon^2}{1 - \mu(A)(2\|x\|_0 - 1)}. \quad (11)$$

Note that as we reconstruct y as one image in A , the smallest error between y and the recovery image is $\mu(A)$, so ε is larger than $1 - \mu(A)$. Therefore,

$$\begin{aligned} \|x^\varepsilon - x\|_2^2 &\leq \frac{4(1 - \mu(A))^2}{1 - \mu(A)(1 + (1/\mu(A)) - 1)} \\ &= 4[1 - \mu(A)]^2 < 4\varepsilon^2. \end{aligned} \quad (12)$$

□

4. Processing for Images under Random Corruption and Contiguous Occlusion

To almost all sparse representation algorithms, as based on errors of pixel in the corresponding position, it is brittle for them to cope with samples under large level of occlusion or corruption [7, 16].

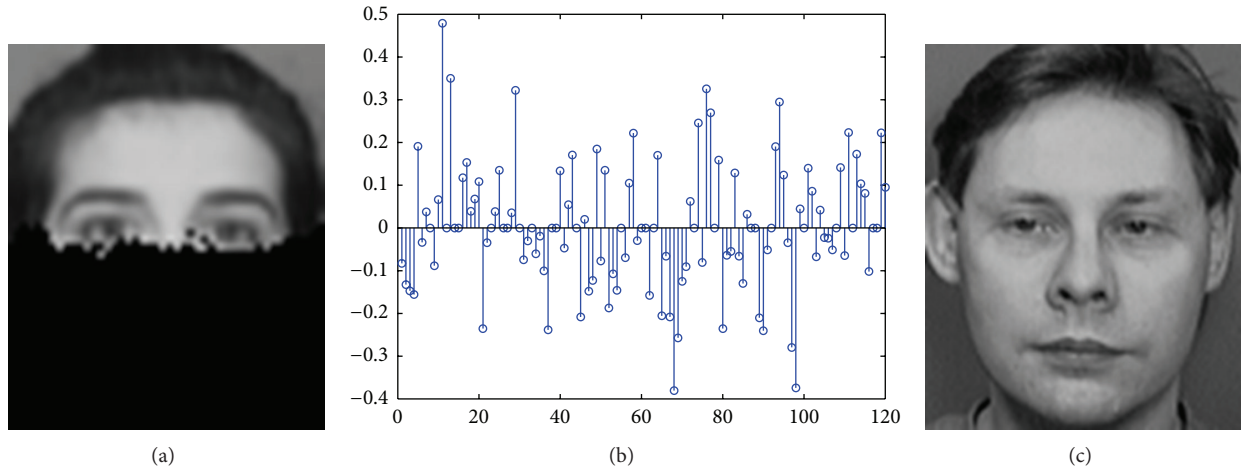


FIGURE 1: (a) is an input sample with contiguous occlusion. (b) is the sparse solution based on the OMP algorithm. (c) is the reconstructed image, whose class is the same as the class of the largest entry of the sparse solution. Obviously, the solution is not sparse enough and we have got a wrong answer due to the occluded part in y .

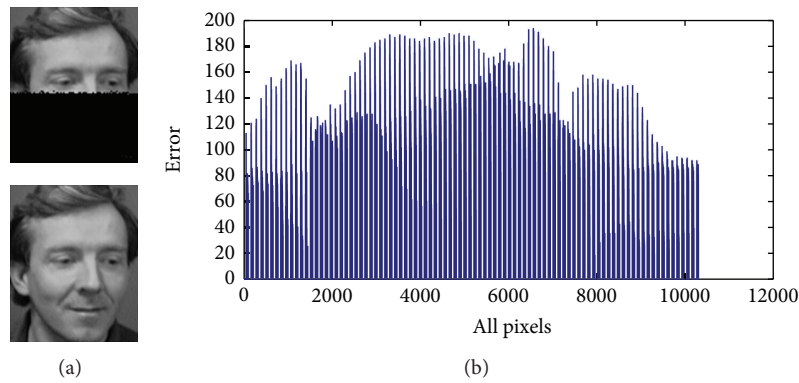


FIGURE 2: Error between an occluded image and a “clean” one. We reshape the pixels in the two samples (a) into vectors and calculate the difference (b). Obviously, the distribution of error is like a comb, which indicates that the error exists in a few pixels and the others are “clean.”

4.1. Necessity of Preprocessing of Test Samples and Training Dictionary. When the test sample y is occluded seriously, the error between y and each column of A increases, which means each element of vector e rises, where $e_i = y - a_i$. Hence, both the largest coefficient and the sparse level after some fix iteration times can hardly reach their respective predetermined thresholds. On condition that we employ A and y directly, as shown in Figure 1, the solution we get can be dense, which will surely weaken the superiority of sparse representation. Therefore it is necessary to process input sample y and training dictionary A firstly.

It has been proposed that this kind of issue can be solved by block partitioning, which means to partition the image into blocks and process each block independently. The results for individual blocks are then aggregated. This method is only valid to images under contiguous occlusion. And the processing takes quite a long time since it transforms the classification to several subproblems. Let us think about how human brains handle the face images disguised by scarf or glasses. We make out the scarf or glasses in the image and then neglect these parts which are unrelated to face. And

our judgment of the person in the image depends on the other parts which we regard as face. Imitating the method human brain dealing with this kind of image, we can propose to preprocess the input image and training matrix before applying them to the OMP algorithm. No matter the random corruption or the contiguous occlusion part can be regarded as noise in the image, we can uniformly reject this part and only pay attention to the other parts.

4.2. How to Identify a Corrupted or Occluded Test Sample Automatically. Firstly, the system should automatically identify whether the test image is the one with partial occlusion or corruption. The errors between the occlusion and corruption image which belong to a class in the training dictionary or an invalid one are both quite large. But the error of an occluded one has some distinguishing features—the error between pixels focuses on only a few pixels; the others' error is quite small. So if the error vector contained some elements which are closed to 0 and the variance of the error vector is

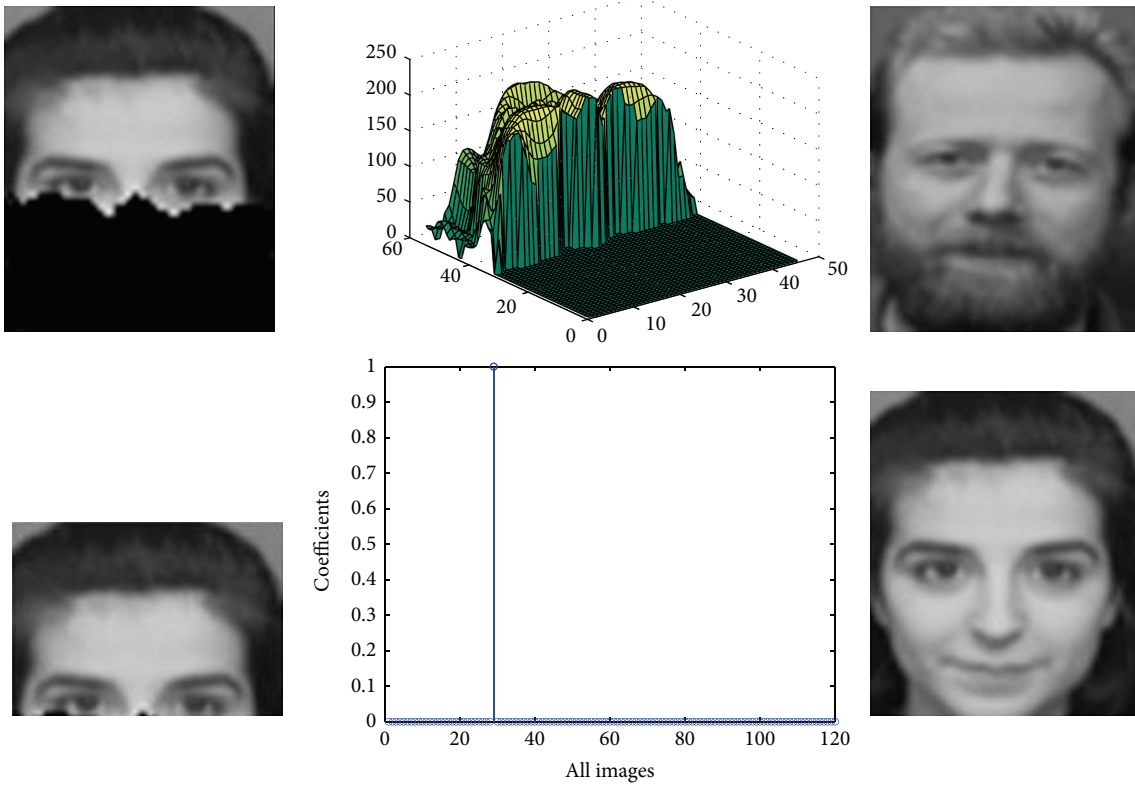


FIGURE 3: Applying the occluded image (left top) directly to the algorithm, we get the wrong answer because part of the noise is too large for the system. By rejecting some pixels in the test image and shrinking the training matrix, respectively, the right person can be recognized.

large enough, $\sigma_e > \sigma_0$, we can regard the test sample as a partial occlusion or corruption one, as shown in Figure 2.

4.3. Images Preprocessing to Corrupted or Occluded Samples. Then let us discuss how to use extract the “clean” pixels and remove the invalid ones. To begin with, we can define an error threshold ξ between a particular pixel in test sample y and respective pixels of elements in A . If the minimal error between the pixel in y and respective pixel in one class of A is larger than ξ ,

$$\min(y_i - a_{ij}) > \xi. \quad (13)$$

We can regard this pixel as an invalid one and remove it from y as well as the respective one from A . Then we get the new training matrix A_1 and test vector y_1 , whose “noise pixels” have been filtered. So the problem has been transformed into finding a sparse solution x_1 subject to

$$y_1 = A_1 \cdot x_1. \quad (14)$$

And the improved OMP algorithm can just be used in this new generated equation. One example is given in Figure 3. This will surely enhance the recognition rate of the system. Since this method does not involve any constraint conditions regarding noise distribution, the issue of random corruption and contiguous occlusion can be solved simultaneously. “Shrinking” of A and y also makes the computing speed faster.

5. Experimental Results

In this section, we apply our algorithm on ORL database for face recognition. We first test the recognition rate and elapsed time of the algorithm, compared to the state-of-art algorithm to find the sparse solution. We then examine the identifying performance to corruption and occlusion. Finally, we simulate the real situation and check up the robustness under various disguises.

5.1. Performance on ORL Database and Yale Database. ORL database consists of 400 frontal images for 40 individuals. Samples in this database include facial variation like various expressions and postures, which can be obstacles or challenges for the system to find the true class of the test images.

We test the face recognition rate and elapsed time of the algorithm by 10-fold cross-validation. In other words, for each test, the training dictionary is consisted of 40 classes of 360 images (9 samples per class) and the remaining 40 images are test samples. All these images have been simply downsampled, without any particular feature extraction. We compared the result with original OMP, together with some state-of-art algorithms aiming at sparse solution—primal augmented Lagrangian method (PALM), dual augmented lagrangian method (DALM), fast iterative soft-thresholding algorithm (FISTA), and truncated Newton interior-point Method (TNIPM). All results of each test and the average are in Table 1.

TABLE 1: Comparison on ORL database.

Test number	Elapsed time per test image(s)					
	PALM	DALM	FISTA	TNIPM	OMP	Improved OMP
1	1.0957	2.6292	0.3548	0.9861	33.1391	0.0278
2	1.3422	2.6434	0.3628	1.0106	35.4203	0.0248
3	1.5197	2.6408	0.3607	0.9211	34.7244	0.0233
4	1.5638	2.6922	0.3661	0.9212	35.2557	0.0253
5	1.2596	2.6749	0.3655	0.8308	36.5687	0.0238
6	1.4026	2.6485	0.3635	0.8343	36.8514	0.0259
7	1.3802	2.7209	0.3682	0.8390	36.7581	0.0218
8	1.2477	2.7298	0.3742	1.0580	36.8528	0.0208
9	1.4948	2.7602	0.3635	0.8528	35.9073	0.0213
10	1.3220	2.6478	0.3670	1.04	38.6038	0.0214
Average(s)	1.3628	2.6788	0.3646	0.9294	36.008	0.0236
Test number	Recognition rate					
	PALM	DALM	FISTA	TNIPM	OMP	Improved OMP
1	62.5%	87.5%	90%	87.5%	77.5%	95%
2	70%	90%	90%	90%	82.5%	92.5%
3	77.5%	92.5%	92.5%	92.5%	80%	95%
4	80%	90%	92.5%	90%	80%	95%
5	70%	92.5%	92.5%	92.5%	85%	97.5%
6	77.5%	92.5%	92.5%	92.5%	82.5%	95%
7	75%	92.5%	95%	95%	87.5%	95%
8	67.5%	92.5%	92.5%	92.5%	85%	97.5%
9	82.5%	95%	95%	95%	85%	97.5%
10	75%	92.5%	95%	92.5%	82.5%	95%
Average	73.75%	91.75%	92.75%	92%	82.75%	95.5%

TABLE 2: Comparison on Yale database.

Test number	Elapsed time per test image(s)					
	PALM	DALM	FISTA	TNIPM	OMP	Improved OMP
1	1.9440	1.3520	5.9824	20.1194	12.433	0.7253
2	1.6345	1.3967	6.4434	17.1639	11.1681	0.7690
3	1.8704	6.3440	6.3742	17.4002	11.3312	0.7831
4	1.9500	2.0409	7.0701	18.1936	11.0440	0.7920
5	1.6420	1.3959	6.1629	16.8426	10.5832	0.7215
6	1.3926	1.3948	6.1054	15.7372	11.6463	0.7824
7	1.5375	1.3638	7.3222	15.9961	13.0719	0.6903
8	1.3519	1.4371	7.8755	15.6780	11.1364	0.8467
Average(s)	1.6654	2.0907	6.6670	17.1414	11.5518	0.6731
Test number	Recognition rate					
	PALM	DALM	FISTA	TNIPM	OMP	Improved OMP
1	52.96%	47.37%	92.11%	55.92%	78.95%	92.43%
2	53.29%	50.33%	95.39%	90.13%	81.58%	92.76%
3	51.97%	52.63%	92.76%	87.17%	83.22%	90.46%
4	53.29%	51.64%	92.43%	91.12%	85.53%	95.72%
5	47.37%	49.34%	94.41%	91.78%	79.93%	94.74 %
6	50%	51.64%	96.38%	93.75%	82.89%	96.71%
7	49.34%	47.37%	93.42%	92.76%	82.89%	92.43%
8	51.64%	50%	95.39%	94.74%	86.51%	95.72%
Average	51.23%	50.04%	94.04%	87.17%	82.69%	93.87%

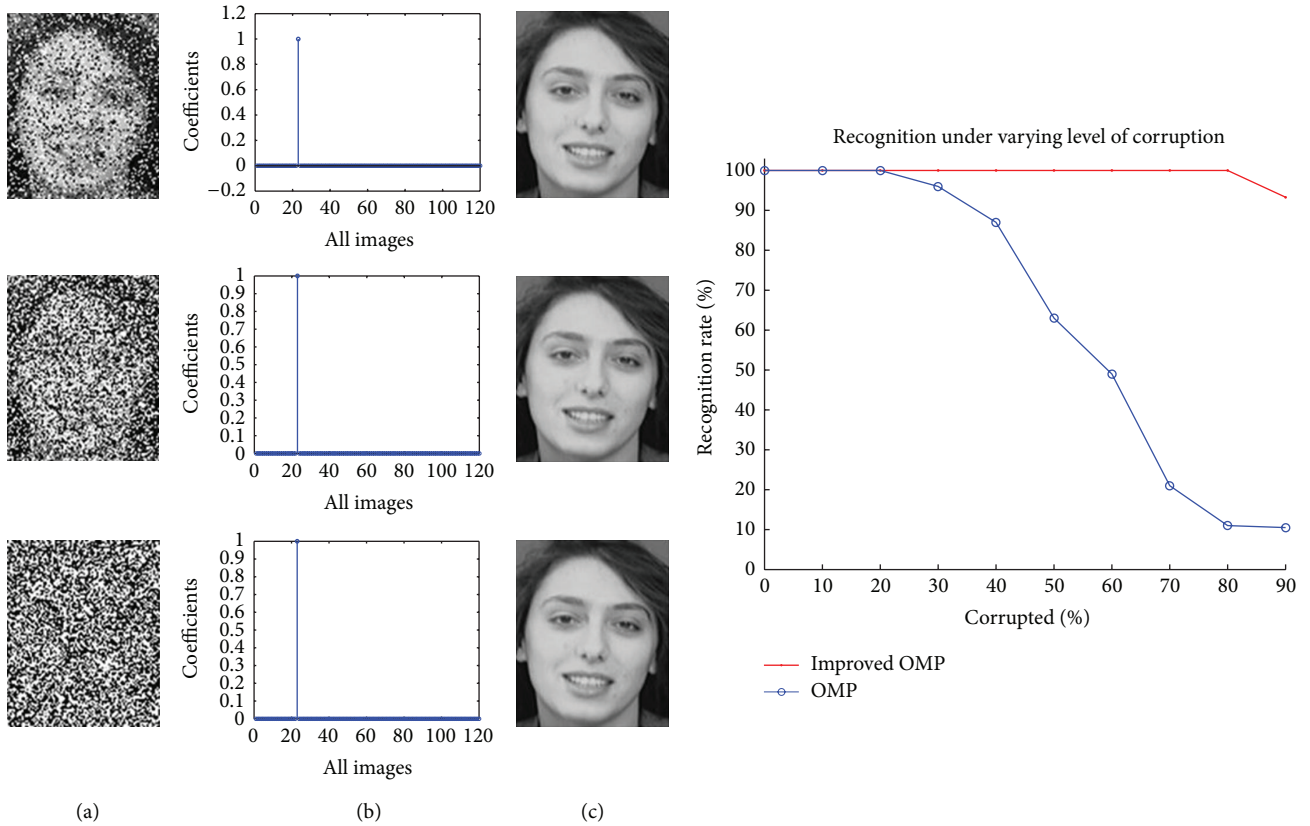


FIGURE 4: Recognition under random corruption. Left: (a) test images y from ORL database, with random corruption. Top row: 30% of pixels are corrupted, middle row: 60% corrupted, bottom row: 90% corrupted. (b) Estimated sparse coefficients. (c) Reconstructed images. We can see that our algorithm outperforms the original OMP.

TABLE 3: Comparison on FERET database.

Test number	Elapsed time per test image(s)					
	PALM	DALM	FISTA	TNIPM	OMP	Improved OMP
1	13.6277	0.6245	3.7790	8.8048	10.1515	0.3854
2	13.6254	0.6191	3.8656	9.4325	10.4445	0.3325
3	18.9904	0.6191	3.8538	8.6242	10.2513	0.2858
4	15.2641	0.6316	4.0842	9.0337	10.2262	0.3229
5	15.2024	0.6364	4.0117	9.0880	10.4940	0.4109
6	13.1985	0.6125	3.8448	8.7168	10.2518	0.4428
7	14.5053	0.6365	3.7359	8.8999	10.8678	0.2599
Average(s)	14.9163	0.6257	3.8821	8.9428	10.3839	0.3486
Test number	Recognition rate					
	PALM	DALM	FISTA	TNIPM	OMP	Improved OMP
1	40%	90.67%	92.67%	90.67%	51.33%	93.33%
2	48.67%	88%	92.67%	88%	50%	92%
3	45.33%	65.33%	68.67%	65.33%	77.33%	92.67%
4	38.67%	84.67%	88%	84.67%	72.67%	90.67%
5	44.67%	86.67%	92%	86.67%	74.67%	93.33%
6	54.67%	80%	86%	80%	57.33%	89.33%
7	42.67%	90%	93.33%	90%	64%	94.67%
Average	44.95%	83.62%	87.62%	83.62%	63.90%	92.29%

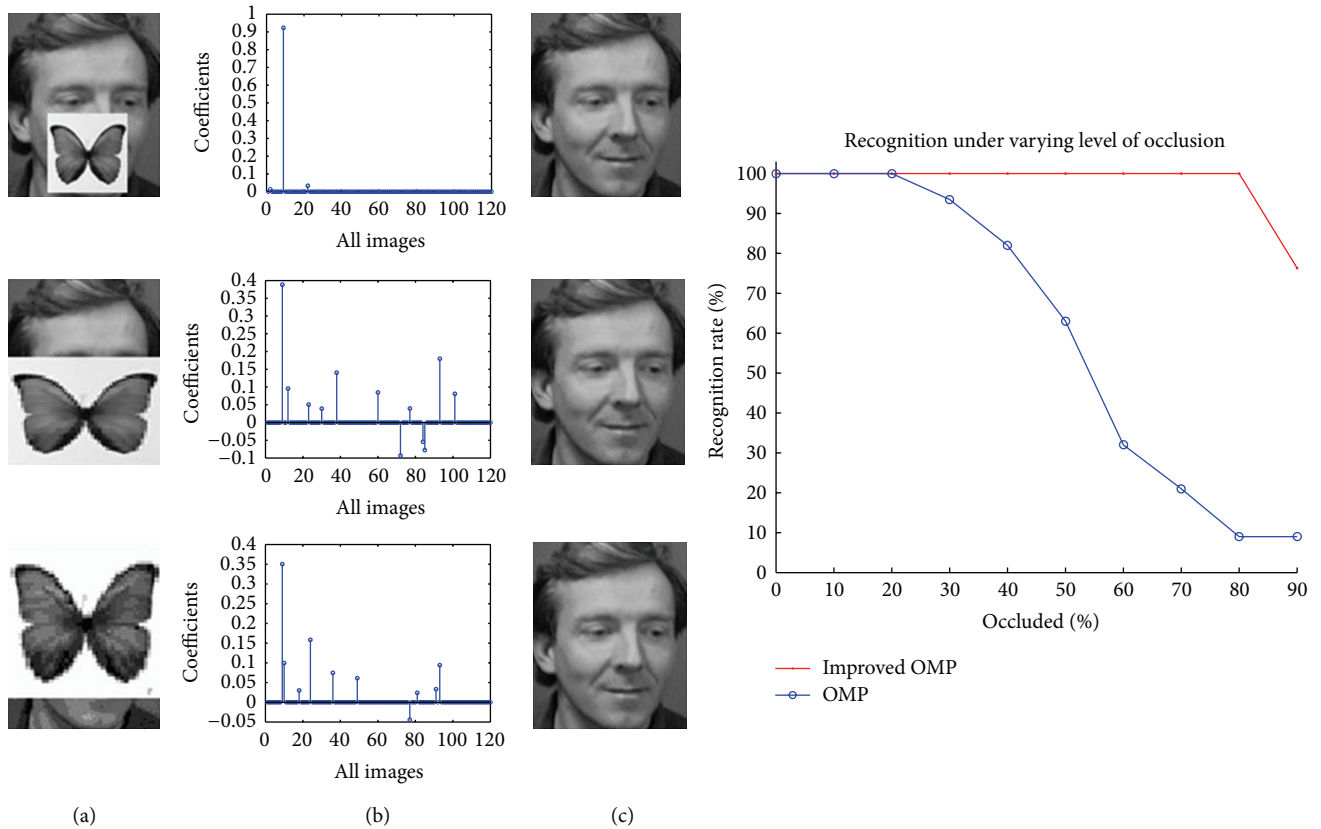


FIGURE 5: Recognition under varying levels of contiguous occlusion. (a) Top row: 30% occluded test sample, middle row: 60% occluded, and bottom row: 90% occluded. (b) Estimated sparse coefficients. (c) Reconstructed images.

As shown in Table 1, the recognition rate of improved OMP is the highest. Meanwhile, elapsed time per sample outperforms others.

We also compared the algorithm with others in the Yale database. Yale database contains 2432 frontal images for 38 individuals, which were captured under various laboratory-controlled lighting conditions. 8-fold cross-validation has been taken in this database, with 56 images of each class as training samples, and the other 304 images as test samples. Just as the tests on ORL database, the images have only been downsampled to construct the training dictionary to make the problem $Ax = y$ become undetermined. The results of the 8 tests and average values are in Table 2.

In Table 2, although the recognition rate of FISTA is higher than our algorithm, its run time is almost tenfold compared with the improved OMP.

Similar experiments were performed on FERET database. Compared with the other face databases we mentioned above, this database includes more variations like postures and facial expressions. We ran 7-fold cross-validation, with 150 classes (6 samples per class) in this database as training database, the other 150 samples as test samples in each test. Data in Table 3 reflects the comparison of these algorithms.

In Table 3, we can see that the average recognition rate of improved OMP is the highest in tests of FERET database and its run time is much shorter.

We get these data in MATLAB on a typical 2.40 GHz PC with quad-core processor. To be fair, both the training dictionary and the test samples are the same to all algorithms. And all identified results depend on the class of the largest coefficient in the sparse solution. The improved OMP algorithm greatly reduces the run time, and its recognition rates are relatively well.

5.2. Recognition despite Occlusion and Corruption

5.2.1. Recognition under Random Corruption.

We first test the robust version of our algorithm for samples under random corruption. We add salt and pepper noise of different intensities to samples in the database to generate the test samples. Figure 4 plots the recognition results of the robust version of the OMP and applying OMP directly to the test samples.

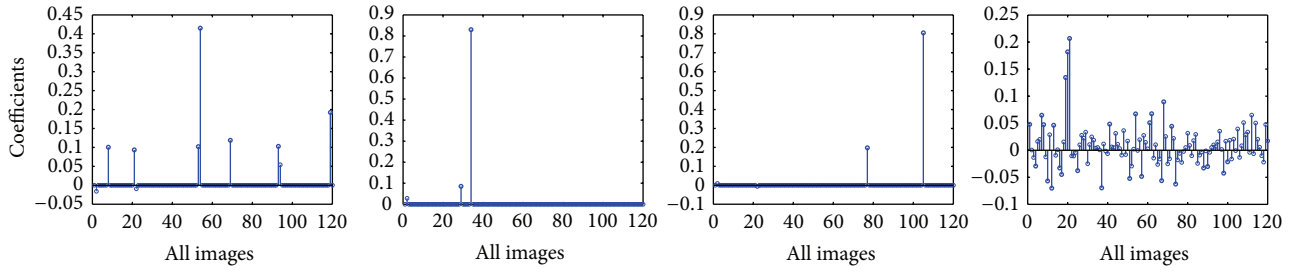
From Figure 4, when the face imaged is 90% corrupted by the noise, although we can hardly identify it as a face image, the algorithm still reconstructed the right image. The right line graph in Figure 4 indicates that this method performs quite well under the condition of large percent of random corruption.

5.2.2. Recognition under Continuous Occlusion.

We add an irrelevant image with different sizes to the samples in the



(a)



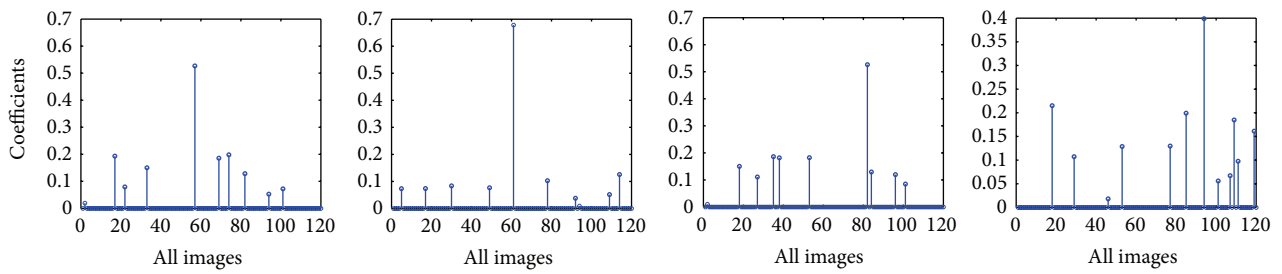
(b)



(c)

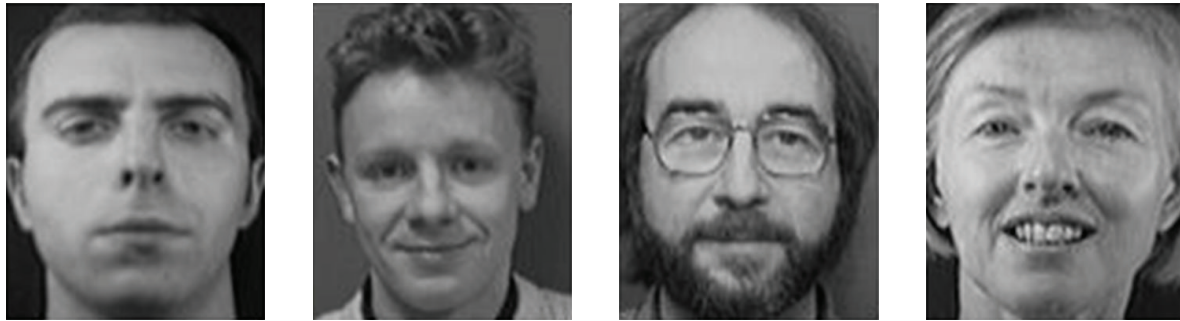


(d)



(e)

FIGURE 6: Continued.



(f)

FIGURE 6: Recognition despite disguise like sunglasses, scarf, or respirator. (a) Test images with sunglasses or scarf. (b) Estimated sparse coefficients. (c) Reconstructed images. (d) Test images with sunglasses and scarf. (e) Estimated sparse coefficients. (f) Reconstructed images.

database and treat them as test image to test the robustness under continuous occlusion. Figure 5 indicates the performance.

We can see in the right of Figure 5 that the improved OMP significantly outperforms the original one, which shows its robustness of occlusion.

5.3. Recognition despite Disguise. Face photos taken in a real world scenario often contained glasses and scarf, which makes it harder for the system to identify the right person. Now let us examine the performance of the algorithm under these kinds of situation. Our test images are also from ORL database and we add glasses and scarf pictures to the samples.

Figure 6 shows that the algorithm also has a quite well performance on the real situation—disguised test samples. We constructed a disguised test sample database based on ORL database of 40 samples with sunglasses and scarf. And the recognition rate reached 95%.

6. Conclusions

In this paper, we proposed a fast and robust algorithm based on OMP algorithm. We first discussed the disadvantage of OMP algorithm to solve the face recognition problem. Then an improved method is proposed to make the elapsed time become much shorter to identify a test image. We also tried to enhance robustness to the occluded and corrupted test samples by extracting the “noiseless” pixel and reduce the elements in both the test image and training dictionary, respectively. Finally, we prove this method by experiments on ORL database, Yale database, and FERET database.

One further work is to enhance the robustness in situation under various kinds of misalignment and postures. We may further reduce the constrained condition and apply this method to object recognition.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (no. 61379010) and Natural Science Basic Research Plan in Shaanxi Province of China (no. 2012JQ1012).

References

- [1] J. Wright, A. Yang, A. Ganesh et al., “Robust face recognition via sparse representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.
- [2] Y. Su, S. G. Shan, X. L. Chen et al., “Hierarchical ensemble of global and local classifier for face recognition,” *IEEE Transactions on Image Processing*, vol. 9, no. 2, pp. 273–292, 2009.
- [3] A. Yang, A. Ganesh, S. Sastry, and Y. Ma, “Fast l_1 -minimization algorithms and an application in robust face recognition: a review,” in *Proceedings of the International Conference on Image Processing*, 2010.
- [4] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan, “Sparse representation for computer vision and pattern recognition,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, 2010.
- [5] E. J. Candès, J. K. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.
- [6] E. J. Candès, J. K. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.
- [7] A. Martinez, “Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 748–763, 2002.
- [8] D. Donolo and M. Elad, “Optimal sparse representation in general(nonorthogonal) dictionaries via minimization,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 5, pp. 2197–2202, 2003.
- [9] D. L. Donoho, “For most large underdetermined systems of linear equations the minimal l_1 -norm solution is also the sparsest solution,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

- [10] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.
- [11] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.
- [12] A. Yang, M. Gastpar, R. Bajcsy, and S. Sastry, "Distributed sensor perception via sparse representation," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1077–1088, 2010.
- [13] D. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18914–18919, 2009.
- [14] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.
- [15] F. Sanja, D. Skocaj, and A. Leonardis, "Combining reconstructive and discriminative subspace methods for robust classification and regression by subsampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 3, 2006.
- [16] J. Wright and Y. Ma, "Dense error correction via ℓ^1 -minimization," *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3540–3560, 2010.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

