A Fast Extension for Sparse Representation on Robust Face Recognition

Hui-ning Qiu
Department of Mathematics
Sun Yat-sen University
Guangzhou, China
qiuhuining@gmail.com

Duc-Son Pham
Department of Computing
Curtin University
Perth, Australia
dspham@ieee.org

Svetha Venkatesh
Department of Computing
Curtin University
Perth, Australia
svetha@cs.curtin.edu.au

Wan-quan Liu
Department of Computing
Curtin University
Perth, Australia
w.liu@curtin.edu.au

Jian-huang Lai
Department of Automation
Sun Yat-sen University
Guangzhou, China
stsljh@mail.sysu.edu.cn

***Abstract*—We extend a recent Sparse Representation-based Classification (SRC) algorithm for face recognition to work on 2D images directly, aiming to reduce the computational complexity whilst still maintaining performance. Our contributions include: (1) a new 2D extension of SRC algorithm; (2) an incremental computing procedure which can reduce the eigen decomposition expense of each 2D-SRC for sequential input data; and (3) extensive numerical studies to validate the proposed methods.**

***Keywords*-Compressive Sensing; Sparse Representation; Face Recognition; Incremental Learning**

I. Introduction

Over the last decade, there has been rapid development of face recognition technology, and many algorithms have been proposed, such as Eigenface [8], Fisherface [2], subspace learning [3], [4]. In a recent work, Wright *et al.* [9] proposed a Sparse Representation-based Classification (SRC) algorithm for robust face recognition, which is inspired from the lately developed Compressive Sensing (CS) theory, and this could help with the occlusion face recognition problem. However, the computation cost of this SRC algorithm is very expensive when the image size is large, thus the algorithm is restricted only to small images.

In this paper we introduce a new formulation of the SRC model working directly on 2D images. Further, we consider applying the SRC algorithm in an incremental learning context, and propose an incremental 2D-SRC learning procedure which can be more efficient.

This paper is organized as follows. In Section II we review the SRC algorithm and propose the 2D-SRC algorithm as well as the incremental computing procedure. In Section III we numerically demonstrate the advantages of 2D-SRC versus 1D-SRC over typical face datasets. Concluding remarks are given in Section IV.

II. The SRC Algorithm

The detail of the original SRC algorithm is reported in [9]. Here, we only briefly describe its major aspects. Essentially, it uses all known training face images to span a face subspace, and for an unknown face image it tries to reconstruct the image sparsely.

The motivation of this model is that given sufficient training samples of each person, any new (test) sample for this same person will approximately lie in the linear span of the training samples associated with the person. To be more precise, let us say, database $\mathcal{A}$ consists of k classes $\{\nu_{1,1},\ldots,\nu_{1,n_1};\ldots;\nu_{k,1},\ldots,\nu_{k,n_k}\}$, where $\nu_{i,l}$ is the l-th image of class i, by stacking pixels of each image into a column vector $\mathbf{v}_{i,l}$, we can build up a matrix A to hold the N training samples

$$A=[\mathbf{v}_{1,1},\ldots,\mathbf{v}_{1,n_1},\ldots,\mathbf{v}_{k,1},\ldots,\mathbf{v}_{k,n_k}]\in\Re^{L\times N} \quad (1)$$

where $L=wh$ is the pixels count of an $h\times w$ image.

Once a new test image $\mathbf{y}$ is acquired, it can be represented using samples from the database

$$\mathbf{y}=\mathbf{A}\mathbf{x}_0 \quad (2)$$

According to the assumption that an individual's given images are sufficient to represent themselves, the solution $\mathbf{x}_0$ in linear equation (2) should be very sparse. This leads to solving a noise-aware ℓ^1-minimization problem

$$(\ell_s^1):\hat{\mathbf{x}}=\arg\min_{\mathbf{x}}\|\mathbf{x}\|_1 \quad \text{s.t. } \|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2\le\epsilon. \quad (3)$$

To recognize a probe image $\mathbf{v}$, the SRC algorithm computes per-class reconstructing residuals and identifies it as the class having the minimum residual. The robust performance of the SRC algorithm has been proved experimentally on face datasets with noises and occlusions.

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.256

A. The 2D-SRC Algorithm

The main disadvantage with (3) is that the problem size is proportional to the size of matrix A, i.e. $size(A) = L \times N$, where L is image pixel counts $L = wh$, and N is the training samples count. This computation becomes very expensive and potentially not practical when the image size is very large (typical images have about 10^4 pixels or more) as ℓ^1 solvers have typical *polynomial* complexity.

To address this issue, we begin with a 2D-CS model

$$(\ell_s^1)' : \hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \ \lambda\|\mathbf{x}\|_1 + \|\mathbf{A} - \sum_i x_i \mathbf{A}_i\|_F^2 \qquad (4)$$

Notice that we do not need to convert images into column vectors as in the previous 1D-CS model. By expanding (4) one can easily show that

$$\|\mathbf{A} - \sum_i x_i \mathbf{A}_i\|_F^2 = \mathbf{x}^T \mathbf{Q} \mathbf{x} - 2\mathbf{b}^T \mathbf{x} + c \qquad (5)$$

where

$$\begin{aligned} \mathbf{Q} &= \begin{pmatrix} \langle \mathbf{A}_1, \mathbf{A}_1 \rangle_F & \cdots & \langle \mathbf{A}_1, \mathbf{A}_N \rangle_F \\ \vdots & \ddots & \vdots \\ \langle \mathbf{A}_N, \mathbf{A}_1 \rangle_F & \cdots & \langle \mathbf{A}_N, \mathbf{A}_N \rangle_F \end{pmatrix} \in \Re^{N \times N} \\ \mathbf{b} &= \begin{pmatrix} \langle \mathbf{A}_1, \mathbf{A} \rangle_F \\ \vdots \\ \langle \mathbf{A}_N, \mathbf{A} \rangle_F \end{pmatrix} \in \Re^{N \times 1} \\ c &= \|\mathbf{A}\|_F^2 \end{aligned}$$

and Frobenius inner product is defined as

$$\langle \mathbf{A}, \mathbf{B} \rangle_F = \sum_i \sum_j A_{ij} B_{ij} \qquad (6)$$

This shows that the 2D-CS model objective function is actually a quadratic form, and the problem size is in proportion to the input samples count N. As $\mathbf{Q}$ is symmetric and positive semidefinite, we can always find vector columns $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_N]$ such that

$$\mathbf{Q} = \mathbf{P}^T \mathbf{P} \qquad (7)$$

Once we have this decomposition of $\mathbf{Q}$, then we can rewrite the quadratic form (5) as

$$\begin{aligned} (5) &= (\mathbf{P}\mathbf{x})^T (\mathbf{P}\mathbf{x}) - 2\left(\mathbf{P}^T \mathbf{z}\right)^T \mathbf{x} + c & (8) \\ &= (\mathbf{P}\mathbf{z})^T (\mathbf{P}\mathbf{x}) - 2\mathbf{z}^T (\mathbf{P}\mathbf{x}) + \mathbf{z}^T \mathbf{z} - \mathbf{z}^T \mathbf{z} + c & (9) \\ &= \|\mathbf{P}\mathbf{x} - \mathbf{z}\|^2 - \mathbf{z}^T \mathbf{z} + c & (10) \end{aligned}$$

where $\mathbf{P}^T \mathbf{z} = \mathbf{b}$. Thus, the original 2D-CS problem can be transformed into an equivalent 1D-CS problem

$$(\ell_s^1)'' : \hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \ \lambda\|\mathbf{x}\|_1 + \|\mathbf{P}\mathbf{x} - \mathbf{z}\|_2^2 \qquad (11)$$

The problem size now becomes $size(P) = r \times N$, which is smaller than that of the 1D-CS model since we can require $r \leq \min(L, N)$.

The only question left is how to find suitable $\mathbf{P}$ and $\mathbf{z}$ that satisfy (7)-(10). Generally, there exists infinite possible P satisfying (7), but we prefer the column number of $\mathbf{P}$ to be as small as possible. Luckily, the SVD of $\mathbf{Q}$ provides us with a compact and "economy size" solution

$$\mathbf{Q} = \mathbf{U}\mathbf{S}\mathbf{U}^T = \left(\mathbf{U}\mathbf{S}^{1/2}\right)\left(\mathbf{U}\mathbf{S}^{1/2}\right)^T \qquad (12)$$

It suggests that if we define

$$\mathbf{P} = \left(\mathbf{U}\mathbf{S}^{1/2}\right)^T \qquad (13)$$

then $\mathbf{P}$ is feasible. Furthermore, since the columns of $\mathbf{U}$ is orthogonal, it can be shown that we can easily ensure $\mathbf{P}^T \mathbf{z} = \mathbf{b}$ by selecting

$$\mathbf{z} = \mathbf{S}^{\dagger} \mathbf{P} \mathbf{b} \qquad (14)$$

This completes the solution of the 2D extension of sparse representation model. We should point out that the computation here only involves sample-pairwise inner product values $\langle \cdot, \cdot \rangle_F$, so other inner product based transformation (e.g. kernel tricks) can be combined with this model.

Based on the original 1D-SRC algorithm, we can describe an 2D-SRC algorithm as following:

Algorithm 1: (2D-SRC)

1. Input: a set of k-class training images $\{\mathbf{A}_{1,1}, \ldots, \mathbf{A}_{1,n_1}, \ldots, \mathbf{A}_{k,1}, \ldots, \mathbf{A}_{k,n_k}\} \subset \Re^{h \times w}$, a test sample $\mathbf{A} \in \Re^{h \times w}$.
2.(a) Compute inner products $\mathbf{Q}$ and vector $\mathbf{b}$.
2.(b) Apply SVD on $\mathbf{Q}$ to find $\mathbf{P}$ and $\mathbf{z}$ by (13) and (14).
2.(c) (optional truncating) Remove columns of $\mathbf{P}$ corresponding to the smallest singular values.
3. Solve the noise-aware ℓ^1-minimization (1D-CS) problem

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \ \lambda\|\mathbf{x}\|_1 + \|\mathbf{P}\mathbf{x} - \mathbf{z}\|_2^2 \qquad (15)$$

4. Compute the per-class residuals

$$r_p(\mathbf{A}) = \|\mathbf{A} - \sum_i \delta_p^{(i)}(\hat{\mathbf{x}}) \ A_i\|_F \quad \text{for } p = 1, \ldots, k. \qquad (16)$$

where $\delta_p(\hat{\mathbf{x}})$, for $p = 1, \ldots, k$ is the components selecting vector for the p-th class whose entries are defined as

$$\text{for } i = 1, \ldots, N, \ \delta_p^{(i)}(\hat{\mathbf{x}}) = \begin{cases} \hat{x}_i, & \text{if } A_i \text{ in the } p\text{-th class} \\ 0, & \text{otherwise} \end{cases} \qquad (17)$$

5. Output: $identity(\mathbf{A}) = class(p_*), \ p_* = \arg\min_p r_p(\mathbf{A})$.

B. The Incremental 2D-SRC Algorithm

In the above approach, computing SVD of $\mathbf{Q}$ is an important step. The size of the matrix $\mathbf{Q}$ is proportional to the number of training images. In the case where the input training images arrive sequentially, or when the number of training images N is very large, direct SVD might not be numerically feasible. To further improve the SVD steps of the 2D-SRC algorithm, we propose an incremental version as follows.

Denoted as $\mathbf{Q}_n$ and $\mathbf{Q}_{n+1}$ respectively are the $\mathbf{Q}$ matrix with n and $n+1$ training examples. Given the decomposing results of $\mathbf{Q}_n$, we want to compute the eigenvalue decomposition of $\mathbf{Q}_{n+1}$, which is equivalent to SVD as $\mathbf{Q}$ is an inner product matrix

$$\mathbf{Q}_{n+1} = \begin{bmatrix} \mathbf{Q}_n & \mathbf{b} \\ \mathbf{b}^T & c \end{bmatrix} = \mathbf{U_{n+1}}\mathbf{\Sigma_{n+1}}\mathbf{U_{n+1}}^T \tag{18}$$

where $\mathbf{b} = [\langle \mathbf{A}_i, \mathbf{A}_{n+1} \rangle_F]$, and $c = \langle \mathbf{A}_{n+1}, \mathbf{A}_{n+1} \rangle_F$ are the incremental inner products formed when a new sample A_{n+1} arrives.

The following theorem provides us with the relationship between eigenvalues and eigenvectors of $\mathbf{Q}_n$, $\mathbf{Q}_{n+1}$.

Theorem 1: (Rank-One Incrementally Updating EVD) Given the eigen decomposition

$$\mathbf{Q}_n = \mathbf{U}_n \mathbf{\Sigma_n} \mathbf{U}_n^T \tag{19}$$

where $\mathbf{\Sigma}_n = \text{diag}(\lambda_1, \ldots, \lambda_n)$, $\mathbf{U}_n = [\mathbf{u}_1, \ldots, \mathbf{u}_n]$ are eigenvalues and eigenvectors respectively, then the eigen decomposition of $\mathbf{Q}_{n+1}$ is given by

$$\begin{aligned} \mathbf{Q}_{n+1} &= \begin{bmatrix} \mathbf{Q}_n & \mathbf{b} \\ \mathbf{b}^T & c \end{bmatrix} \equiv \begin{bmatrix} \mathbf{U}_n \mathbf{\Sigma_n} \mathbf{U}_n^T & \mathbf{b} \\ \mathbf{b}^T & c \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{U_n} & 0 \\ 0 & 1 \end{bmatrix} \mathbf{R}_{n+1} \begin{bmatrix} \mathbf{U}_n^T & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{20}$$

where

$$\begin{aligned} \mathbf{R}_{n+1} &= \begin{bmatrix} \mathbf{\Sigma_n} & \mathbf{U}_n^T \mathbf{b} \\ \mathbf{b}^T \mathbf{U}_n & c \end{bmatrix} \equiv \begin{bmatrix} \mathbf{\Sigma_n} & \mathbf{z} \\ \mathbf{z}^T & c \end{bmatrix} \\ &\triangleq \mathbf{V}_{n+1} \mathbf{D_{n+1}} \mathbf{V}_{n+1}^T \end{aligned} \tag{21}$$

as we denote $\mathbf{U}_n^T \mathbf{b} \equiv \mathbf{z} = [z_1, \ldots, z_n]^T$, $\mathbf{D}_{n+1}$ and $\mathbf{V}_{n+1}$ are eigen-pairs of $\mathbf{R}_{n+1}$. $\mathbf{R}_{n+1}$ is called arrowhead matrix and its eigenvalues are exactly $n+1$ roots of the following equation

$$z_1^2/(d-\lambda_1) + \cdots + z_n^2/(d-\lambda_n) = d - c. \tag{22}$$

Moreover, if we let $\lambda_1 < \lambda_2 < \cdots < \lambda_n$ be sorted, and $d_1 < d_2 \cdots < d_n < d_{n+1}$ be all $n+1$ sorted roots of the equation w.r.t. d, then the following interlacing relation holds

$$d_1 < \lambda_1 < d_2 < \cdots < d_n < \lambda_n < d_{n+1} \tag{23}$$

The eigenvector $\mathbf{v}_i$ corresponding to eigenvalue d_i can be obtained by solving the eigen-equation directly, which gives

$$\mathbf{v}_i = \frac{1}{T_i} \left[\frac{z_1}{d_i - \lambda_1}, \cdots, \frac{z_n}{d_i - \lambda_n}, 1 \right] \tag{24}$$

where T_i is a factor to normalize the eigenvector $\mathbf{v}_i$.

The properties of arrowhead matrix shown in Theorem 1 are well studied in literature (e.g. [6], [7]). Next, we design a binary search algorithm to find all $n+1$ eigenvalues of $\mathbf{Q}_{n+1}$ from (22), and then compute their corresponding eigenvectors from (24).

Algorithm 2: (Binary Search for roots of Equation (22))
1. Input: the current n eigenvalues $\lambda_1, \ldots, \lambda_n$, the values $z_1, \ldots, z_i$, and c, $M = norm(Q_n)$
2. For $i = 1, \ldots, n$, binary search d_i in interval $[\lambda_{i-1}, \lambda_i]$ (let $\lambda_{-1} = -M$, $\lambda_{n+1} = M$):
   let $low = \lambda_{i-1}$, $high = \lambda_i$
   do
      $d_i = (\lambda_{i-1} + \lambda_i)$;
      $fval = (d_i - c) - \sum_{i=1}^{n} z_i^2 / (d_i - \lambda_i)$;
      if $fval > 0$ then $low = mid$, else $high = d_i$;
   while $fabs(high - low) > eps$
3. Output: the new $n+1$ eigen-values $d_1, \ldots, d_{n+1}$

Algorithm 3: (Incremental procedure for 2D-SRC)
1. Given: current training images $\{\mathbf{A}_1, \ldots, \mathbf{A}_n\} \subset \Re^{h \times w}$ with their labels set K, a saved copy of SVD decomposition on current $\mathbf{Q}_n = \mathbf{U}_n \mathbf{\Sigma}_n \mathbf{U}_n^T$, a new labeled training image $\mathbf{A}_{n+1} \in \Re^{h \times w}$.
2. Compute $\mathbf{b} = [\langle \mathbf{A}_i, \mathbf{A}_{n+1} \rangle_F]_{i=1,\ldots,n}$, $c = \langle \mathbf{A}_{n+1}, \mathbf{A}_{n+1} \rangle_F$.
3. Compute all eigenvalues $\mathbf{\Sigma_{n+1}} = diag\,(d_1, \ldots, d_{n+1})$ by Algorithm 3.
4. Compute all eigenvectors $\mathbf{U_{n+1}}$ by (24).
5. Continue 2D-SRC procedure (Algorithm 2) with the new base image set $\{\mathbf{A}_1, \ldots, \mathbf{A}_n, \mathbf{A_{n+1}}\}$, but no need to do SVD on $\mathbf{Q}_{n+1}$ separately.
6. $n \leftarrow n+1$, and turn to step 1.

Combining algorithms (1) $\sim$ (3), we can assemble an efficient incremental process which can reuse previous computed results, which is particularly useful in sequence processing and/or incremental learning.

The following table summarizes the computational complexity of three algorithms:

	SVD flops	ℓ^1 problem size
1D-SRC	-	$O(LN)$
2D-SRC	$O(N^3)$	$O(rN)$
2D-SRC+incremental	$O(N^{2+\delta})^*$	$O(rN)$

* (here $0 \leq \delta \leq 1$ is a constant related to matrix-matrix multiplication complexity, which is not well determined in theory yet.)

III. EXPERIMENTS

We have performed experiments on two face databases: the Yale B plus Extended face database [3], [4] and the AR face database [5]. For the Yale B+Extended face database, we only use the frontal faces in subset1 and subset2. From the AR face database we choose 120 individuals and each with 26 images available. We crop and resize them into a size of 90×90, then align them by fixing two eyes points in order to enhance the quality of distributed clusters.

Although many ℓ^1-optimization solvers are publicly available, $l1$-magic [1] is chosen in this paper as the solver for ℓ^1-norm problem, just following [9].

All our results are obtained using MATLAB on a desktop machine with Intel(R) Core(TM)2 CPU, 2GB of RAM hardware configuration.

A. 1D-SRC vs. 2D-SRC Accuracy and Speed

In this part we compare the original 1D-SRC algorithm and our proposed 2D-SRC algorithm under different scaling dimensionality, and we evaluate them in two aspects: the recognition accuracy $R = \frac{num\ of\ correctness}{num\ of\ testing\ samples}$, and the average time expended for recognizing one test sample $T = \frac{accumulated\ time\ expended}{num\ of\ testing\ samples}$. Among all face images, half of them are randomly selected as training images and the others are used for testing. Each image has been resized into several different scales, and then the recognition accuracy R and the average time expended T under all scales are collected in each round. A comparison of results between the 1D-SRC algorithm and the 2D-SRC algorithm is shown in Figs 1 and 2.

We can observe that: (1) the 2D-SRC algorithm can achieve recognition rates similar to the original 1D-SRC algorithm, while they are both effective for face recognition task; (2) but the computational cost of the 2D-SRC algorithm is basically $2 \sim 3$ times faster than that of the original 1D-SRC algorithm. We also notice that as the problem size becomes larger and larger, the average time expended of both algorithms is increasing.

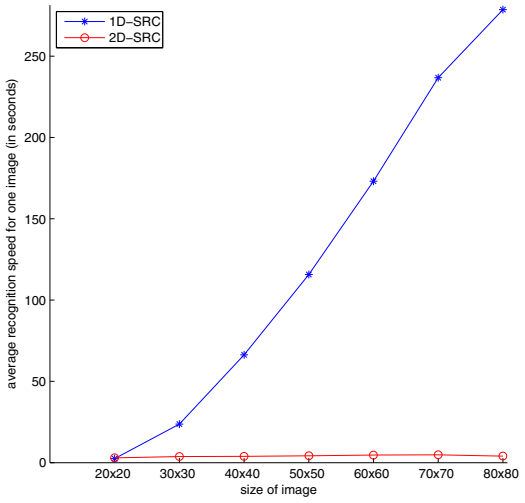
B. Incremental Performance and Efficiency

In this part we conduct schemed performance evaluations (in terms of accuracy and speed) of three algorithms (i.e. 1D-SRC, 2D-SRC, 2D-SRC-incremental) in the context of incremental training process. Among all the images, 60% are used for training, 40% are used for testing; at first we randomly select 50% training samples as a starting training set, then add other training samples one by one, and run recognition tests on a fixed testing set. The average recognition rates and time expended per sample under all training set sizes are recorded and shown in Figs 3 and 4.

We can draw a conclusion that applying incremental procedure on 2D-SRC is helpful to reduce the computational cost further, but that is not as much as the reduction cost from 1D-SRC to 2D-SRC.

	1D-SRC		2D-SRC	
image sizes	avg.rate (%)	avg.time (sec.)	avg.rate (%)	avg.time (sec.)
$siz = 20 \times 20$	100.00	2.5057	100.00	2.9615
$siz = 30 \times 30$	100.00	23.7490	100.00	3.7482
$siz = 40 \times 40$	100.00	66.3346	100.00	3.9022
$siz = 50 \times 50$	100.00	115.7170	100.00	4.2105
$siz = 60 \times 60$	100.00	173.1045	100.00	4.7190
$siz = 70 \times 70$	100.00	236.7434	100.00	4.8268
$siz = 80 \times 80$	100.00	278.6187	100.00	4.0566

(a) comparison of recognition rates

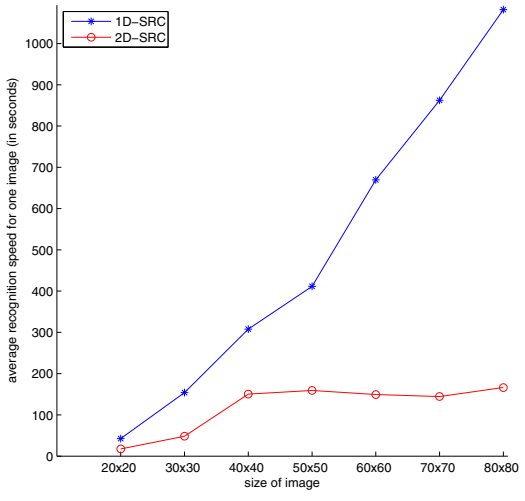


(b) speed vs. dimensionality

Figure 1. Performance on Yale B+Extended Face Database

	1D-SRC		2D-SRC	
image sizes	avg.rate (%)	avg.time (sec.)	avg.rate (%)	avg.time (sec.)
$siz = 20 \times 20$	75.26	42.6049	74.17	17.5065
$siz = 30 \times 30$	75.17	153.9607	74.73	48.3192
$siz = 40 \times 40$	75.65	307.7532	75.03	150.4301
$siz = 50 \times 50$	76.10	411.4108	75.13	159.2576
$siz = 60 \times 60$	76.20	669.4789	75.70	149.4178
$siz = 70 \times 70$	76.45	862.3786	75.90	144.2889
$siz = 80 \times 80$	76.51	1082.0746	76.10	166.2655

(a) comparison of recognition rates

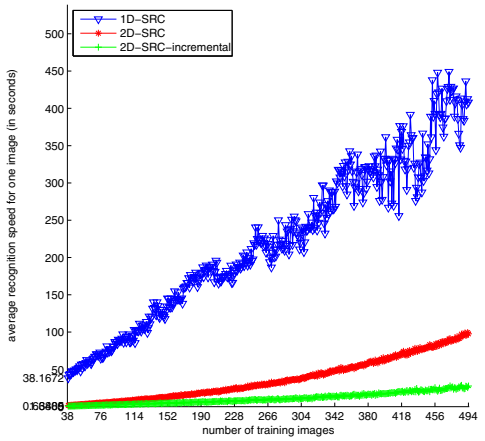


(b) speed vs. dimensionality

Figure 2. Performance on AR Face Database

	1D-SRC	2D-SRC	2D-SRC-incremental
training sizes	avg.rate (%)	avg.rate (%)	avg.rate (%)
$trn = 190$	100.00	100.00	100.00
$trn = 228$	100.00	100.00	100.00
$trn = 266$	100.00	100.00	100.00
$trn = 304$	100.00	100.00	100.00
$trn = 342$	100.00	100.00	100.00
$trn = 380$	100.00	100.00	100.00
$trn = 418$	100.00	100.00	100.00
$trn = 456$	100.00	100.00	100.00
$trn = 494$	100.00	100.00	100.00

(a) comparison of recognition rates

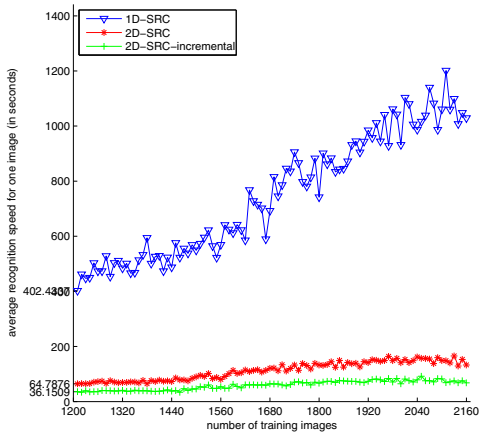


(b) speed vs. train size

Figure 3. Incremental Results on Yale B+Extended Face Database

	1D-SRC	2D-SRC	2D-SRC-incremental
training sizes	avg.rate (%)	avg.rate (%)	avg.rate (%)
$trn = 1200$	75.36	75.10	75.10
$trn = 1320$	75.65	75.38	75.38
$trn = 1440$	76.17	75.87	75.87
$trn = 1560$	76.24	75.96	75.96
$trn = 1680$	78.36	78.03	78.03
$trn = 1800$	79.50	78.76	78.76
$trn = 1920$	80.35	79.95	79.95
$trn = 2040$	82.42	81.86	81.86
$trn = 2160$	84.10	83.90	83.90

(a) comparison of recognition rates



(b) speed vs. train size

Figure 4. Incremental Results on AR Face Database

IV. Conclusions

In this paper we have proposed a faster extension of Wright *et al.*'s sparse representation based classification algorithm for robust face recognition. The proposed method mainly makes use of inner product computation and transformation for acceleration, and it can also work in an incremental way. Experimental results show that the new algorithms still maintain approximate performance at a much faster speed when image size is large.

We agree that even though the proposed techniques are faster than the original method, the overall computing speed of the current SRC algorithms is still relatively slow. This is significant when both the dimensionality of images and the size of base images are large. As a result, further faster solving algorithms and hardware developments are still required, which is our future research direction.

Acknowledgements

This work was supported by the NSF-Guangdong (U0835005), the NSFC (60633030), the 973 Program (2006CB303104), and by the Australian Research Council. The authors would like to thank the reviewers for their comments to improve the quality of the paper.

References

[1] Emmanuel candes and justin romberg, l1-magic: http://www.acm.caltech.edu/l1magic/.

[2] P. N. Belhumeur, J. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE TPAMI*, 19(7):711–720, 1997.

[3] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE TPAMI*, 23(6):643–660, 2001.

[4] K. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE TPAMI*, 27(5):684–698, 2005.

[5] A. Martinez and R. Benavente. The ar face database. Technical report, CVC Tech. Report #24, 1998.

[6] D. P. O'Leary and G. W. Stewart. Computing the eigenvalues and eigenvectors of symmetric arrowhead matrices. *Journal of Computational Physics*, 90(2):497–505, 1990.

[7] L. Shen and B. W. Suter. Bounds for eigenvalues of arrowhead matrices and their applications to hub matrices and wireless communications. *EURASIP Journal on Advances in Signal Processing*, 2009:12, 2009.

[8] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[9] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE TPAMI*, 31(2):210–227, Feburary 2009.