

An Application of a Parallel Algorithm on an Image Recognition

Ran Baik¹

Abstract

This paper is to introduce an application of face recognition algorithm in parallel. We have experiments of 25 images with different motions and simulated the image recognitions; grouping of the image vectors, image normalization, calculating average image vectors, etc. We also discuss an analysis of the related eigen-image vectors and a parallel algorithm. To develop the parallel algorithm, we propose a new type of initial matrices for eigenvalue problem. If A is a symmetric matrix, initial matrices for eigen value problem are investigated: the "optimal" one, which minimize $\|C - A\|_F$ and the "super optimal", which minimize $\|I - C^{-1}A\|_F$. In this paper, we present a general new approach to the design of an initial matrices to solving eigenvalue problem based on the new optimal investigating C with preserving the characteristic of the given matrix A. Fast all resulting can be inverted via fast transform algorithms with $O(N \log N)$ operations.

Key Words: pattern recognition, eigenvalue problem, principle components, initial matrices, Symmetric matrices, Gram-Schmidt.

I. INTRODUCTION

In the pattern recognition system, it is considered that is the characteristic of the vectors on the image. The number of the characteristic vectors that is composed to the dimension of vectors is also very important in the rate of recognition. As the more characteristic factors is, it will be affected weakness on the system. Furthermore, it will be slow to read the learning.

g and recognition rate of the image and we need to the group of the learning for the modeling. The number of the characteristic vectors is getting more i.e. the characteristic vectors that is composed the dimensions of the vectors is higher. On the most of pattern recognition, the capacity of the pattern recognition is better until some range, but it will happen rather decreasing than increasing. By the principle components analysis, what is the optimal reduction from higher dimension of characteristic vectors to lower dimension of characteristic vectors. The Principle component is one of multivariate data processing methods that reduced from high dimension to low dimension. Characteristic Data is presented standard axis as many as

the number of dimensions of the characteristic vectors. The n-dimension is presented by the data of n standard axes. If the dimension is reduced, it will be reduced the number of the standard axes. Therefore, the standard axes of statistics multivariate data is to project on a characteristic vector, it will be reduced. PCA (Principle Component Algorithm) is to rearrange the characteristic vectors and transform a standard axis based on the set of uncorrelated variates. In this paper, we analyze the extractions of the characteristic vectors and the algorithm in the parallel cases.

II. DEVELOPMENT OF IMAGE RECONGNITION'S PARALLEL ALOGORITHM

Why do we need this method to rearrange the characteristic vectors? We need the simple example the concrete the cases. We have a simple example of PCA as below:

Manuscript received December 28, 2017; Revised December 29, 2017; Accepted December 30, 2017. (ID No. JMIS-2017-0058)

Corresponding Author: Ran Baik, #417 Eodeung-daero, Gwangsan-gu, Gwangju, Republic of Korea,
Tel: 82-62-940-5425, E-mail: baik@honam.ac.kr.

¹Department of Computer Engineering, Honam University, GwangJu, Korea, baik@honam.ac.kr

2.1. Distributions and transformations of group of the characteristic vectors

A group of random vectors that the value is below will be distributed by given condition.

- (1) To produce natural vector sets
- (2) To rearrange the given vectors from (1) that average of all vectors are zero.
- (3) To reproduce the vectors that the radius from origin is less than 0.5 and ignore others, i.e. greater than 0.5. We can scatter the data that is composed 2-dimension of 1000 random vectors (Figure 1)

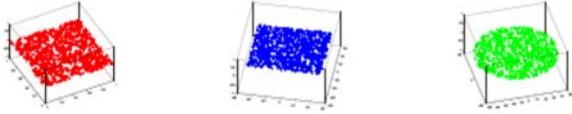


Fig 1. Distribution of set of vectors

Also we consider the step to figure out from x-y coordinates to polar coordinates transformations of the distribution of the given set of the vectors (Fig. 2 and Fig. 3).

In this work, we need to compute the eigenvalues of a covariance matrix to know the principal components of the images. Furthermore, we use the eigenvectors from the eigenvalues to get the information of Principle components.

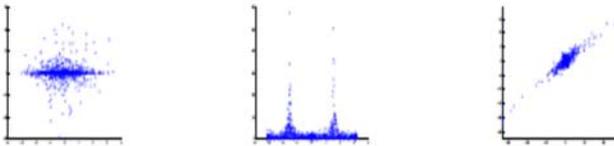


Fig 2. Steps for transformation of the set of vectors.



Fig 3. Principle components of the given vectors.

There are several methods to solve the eigen problem, but there are sequential processes to get all eigenvalues. Also, the structure of the given matrix is broken to do for it. In our work, we want to keep the structure of the given matrix and develop the parallel algorithm. We compare the complexity to compute all eigenvalues of the given matrix between existing algorithm and our algorithm.

2.2. A Construction of an Image Recognition's Algorithm

2.2.1 Development of an algorithm of eigenface on PCA

Each pixel of a face-image is composed of on the face space.

An eigenface is the base vectors which is composed the face space. The base vectors is presented the common characteristics among all candidate face images. The eigen face is produced from difference vectors between an average face image and each candidate face image. These are eigenvectors of the covariance matrix. Since a set of eigenvalues of the covariance matrix is a representation of the range of distribution of average face image, the eigenface which is composed the largest eigenvalue and its eigenvector is the closest face with the origin face. If the eigenvalue is getting smaller, it will decrease the characteristic of the face image. Therefore, we discuss the largest eigenvalue and the smallest eigenvalue of our face image and study the algorithm to compute them in this section.

We apply the algorithm for computing the largest eigenvalue and the smallest eigenvalue independently. We compare the difference between the largest eigenface and the smallest eigenface. We discuss analysis of the algorithm in parallel. First of all, we develop a method to compute the extreme eigenvalues. It is modified by Newton method with a parameter t . It is a procedure to develop the algorithm as follows;

We denote by M_n the space of n -by- n symmetric matrices. We denote by $\sigma(A)$ the set of eigenvalues of $A \in M_n$.

Let $A \in M_n$ be a symmetric matrix. Then there is an orthogonality $U = [U_1, \dots, U_n] \in M_n$ such that

$$A = U \begin{bmatrix} \lambda_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_n \end{bmatrix} U^T, \lambda_k \in \mathbb{R}.$$

The following is the usual Newton method for obtaining an eigenpair of a Symmetric matrix $A \in M_n$ [3]:

Consider $G: \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n \times \mathbb{R}$ such that

$$G \begin{pmatrix} X \\ \alpha \end{pmatrix} = \begin{bmatrix} (\alpha I - A)X \\ X^T X - 1 \end{bmatrix} \quad (2)$$

Then L is the set of solution for $G \begin{pmatrix} X \\ \alpha \end{pmatrix} = 0$.

Assuming the matrix

$$\begin{bmatrix} \alpha I - A & X \\ 2X^T & 0 \end{bmatrix} \in M_{n+1}$$

is invertible, then the usual Newton iteration is

$$\begin{bmatrix} X' \\ \alpha' \end{bmatrix} = \begin{bmatrix} X \\ \alpha \end{bmatrix} - \begin{bmatrix} \alpha I - A & X \\ 2X^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} (\alpha I - A)X \\ X^T X - 1 \end{bmatrix}$$

It is well known that the Newton's method has a local quadratic convergence rate [3], that is there is a small neighborhood N_{ϵ_k} for each eigenpair $\begin{bmatrix} U_k \\ \lambda_k \end{bmatrix}$ such that

if $\begin{bmatrix} X \\ \alpha \end{bmatrix} \equiv \begin{bmatrix} X^{(0)} \\ \alpha^{(0)} \end{bmatrix} \in N_{\epsilon_k}$ then

$$\left\| \begin{bmatrix} X^{(i+1)} \\ \alpha^{(i+1)} \end{bmatrix} - \begin{bmatrix} U_k \\ \lambda_k \end{bmatrix} \right\|_2^2 \leq C \left\| \begin{bmatrix} X^{(i)} \\ \alpha^{(i)} \end{bmatrix} - \begin{bmatrix} U_k \\ \lambda_k \end{bmatrix} \right\|_2^2$$

for all $i = 0, 1, \dots$ where $C < \infty$ is a positive constant.

We call N_{ϵ_k} the quadratic convergence neighborhood of

the eigenpair $\begin{bmatrix} U_k \\ \lambda_k \end{bmatrix}$. Although the specific determination of each N_{ϵ_k} is an extremely difficult task, if the method converges to a point in L then we know the rate of convergence will eventually be quadratic. It can be shown easily that the Newton's method is not global.

We modify the Newton method in order to have a global convergence [2]. There are several considerations to give for the Modification.

First, under the modification we desire the pair $\begin{bmatrix} X^{(i)} \\ \alpha^{(i)} \end{bmatrix}$ gets closer to an eigenpair at each step of the iteration, i.e., $d_L\left(\begin{bmatrix} X^{(i+1)} \\ \alpha^{(i+1)} \end{bmatrix}\right) \leq d_L\left(\begin{bmatrix} X^{(i)} \\ \alpha^{(i)} \end{bmatrix}\right)$ where d_L is a suitable distance measure from a point to L . It will ensure the points under the iteration remain in D . Second, we want to modify the method the least amount as possible in order to preserve the original properties of the Newton's Method, for example, local quadratic convergence. Third, the modified method should be simple to implement, requires almost the same procedures as the original Newton iteration.

2.3 Newton Iteration and Its Modification

Consider the usual Newton iteration (2):

$$\begin{bmatrix} X' \\ \alpha' \end{bmatrix} = \begin{bmatrix} X \\ \alpha \end{bmatrix} - \begin{bmatrix} \alpha I - A & X \\ 2X^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} (\alpha I - A)X \\ X^T X - 1 \end{bmatrix}$$

Choose a parameter $t > 0$ so that the method takes the form

$$\begin{bmatrix} \alpha I - A & X \\ 2X^T & 0 \end{bmatrix} \begin{bmatrix} X' \\ \alpha' \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & t \end{bmatrix} \begin{bmatrix} 0 & X \\ X^T & 1/\alpha \end{bmatrix} \begin{bmatrix} X \\ \alpha \end{bmatrix} \quad (3)$$

$$\text{Then } (\alpha I - A)X' + \alpha X = \alpha' X = \alpha X \quad (4)$$

$$\text{and } 2X^T X' = t(2X^T X + 1) \quad (5)$$

Then the parameterized Newton method takes a form:

$$X' = \frac{1}{\beta} (\alpha I - A)^{-1} X, \quad (6)$$

$$\text{and } \alpha' = \alpha - \frac{t}{\beta}. \quad (7)$$

Now, suppose $\begin{bmatrix} X \\ \alpha \end{bmatrix} \in D$ and let $L = \left\{ \begin{bmatrix} U_k \\ \lambda_k \end{bmatrix} \mid U_k \in R^n, \|U_k\|_2 = 1, \text{ and } \lambda_k \in R \right\}$ be the set of all eigenpairs of A . Define a distance measure from a point $\begin{bmatrix} X \\ \alpha \end{bmatrix}$ to L by

$$d_L\left(\begin{bmatrix} X^{(i+1)} \\ \alpha^{(i+1)} \end{bmatrix}\right) \equiv \|(\alpha I - A)X\|_2 \quad (8)$$

Clearly, $d_L\left(\begin{bmatrix} X \\ \alpha \end{bmatrix}\right) \geq 0$, $d_L\left(\begin{bmatrix} X \\ \alpha \end{bmatrix}\right) = 0$ implies $\begin{bmatrix} X \\ \alpha \end{bmatrix} \in L$, and $d_L: D \rightarrow R^+$ is continuous (since D is compact, d_L is actually uniformly continuous).

We have the following.

Lemma 1[4]: Let $A \in M_n$ be Symmetric. Consider the parameterized Newton's method

$$X' = \frac{1}{\beta} (\alpha I - A)^{-1} X, \text{ and } \alpha' = \alpha - \frac{t}{\beta}, \text{ where}$$

$$\beta = X^T (\alpha I - A)^{-1} X, \text{ and}$$

$$\tilde{\beta} = \|(\alpha I - A)^{-1} X\|_2 = (X^T (\alpha I - A)^{-2} X)^{1/2}.$$

Then $d_L\left(\begin{bmatrix} X' \\ \alpha' \end{bmatrix}\right)$ is minimized at $t = \left(\frac{\beta}{\tilde{\beta}}\right)^2$ with

$$\min_t d_L\left(\begin{bmatrix} X' \\ \alpha' \end{bmatrix}\right) = \frac{1}{\tilde{\beta}} \left(1 - \left(\frac{\beta}{\tilde{\beta}}\right)^2\right)^{1/2}.$$

Therefore, we have the following modification of the Newton's method:

$$X' = \frac{1}{\beta} (\alpha I - A)^{-1} X \quad (9)$$

$$\alpha' = \alpha - \frac{\beta}{\tilde{\beta}^2}. \quad (10)$$

In this 2.3, we discuss it about an application of eigenfaces from 25 different expressions of a face.

2.4 An application of the face recognition

There are 5 steps to construct for face images.

Step 1: An input of the origin images

The size of face image is $N \times N$ and the number of the face images that are recognized as candidates are M such that each is $N^2 \times 1$ column vector. Denote the set of face vectors that are recognized as candidates by S (Fig 4).

$$S = \{I_1, I_2, I_3, \dots, I_M\} \quad (11)$$



Fig 4. Recognition candidate face DB.

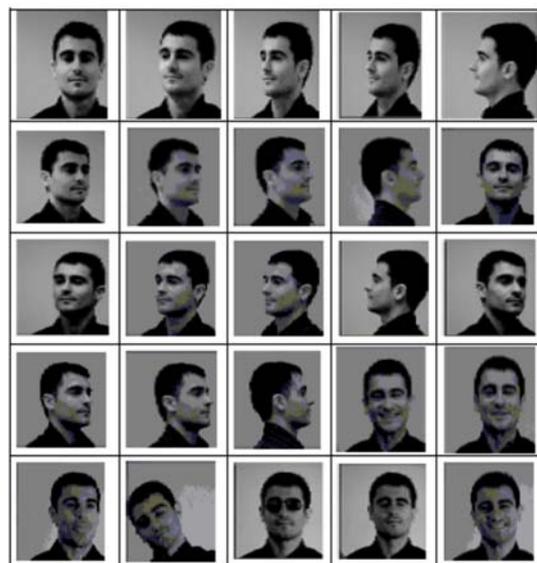


Fig 5. Normalized Recognition candidate face DB.

Step 2: Normalization of the image

We normalize the image on the basis of setting average and variance to reduce errors caused by the light and background (Figure 5).

Step 3: Computation of the average Face's vector

We compute the average face vector from the face set S of recognition candidates (Figure 6).

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n \quad (12)$$



Fig 6. Average face

In the computation for eigenpairs we develop the parallel algorithm with modified Newton method each eigenpair.

Even though, the given matrix is ill conditioned, we can get all eigenpairs. This is an ill-conditioned matrix case:

Example1: Let $H = \begin{bmatrix} 1 & 1/2 & 1/3 & \dots & 1/n \\ 1/2 & 1/3 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 1/n & \dots & \dots & \dots & \frac{1}{2n-1} \end{bmatrix}$ be a

n by n Hilbert matrix H . The Hilbert matrix is a well-known example of an ill-conditioned positive. Because the smallest eigenvalue λ_{12} of $H \in M_{12}$ is so near zero, many conventional algorithms produce $\lambda_{12} = 0$. Our method gives the following experimental results:

Set the convergence criterion, $\varepsilon = 2 \times 10^{-16}$, i.e.,

$$\|(H - \alpha_k I)X_k\|_2 < \varepsilon.$$

Suppose $D = \left\{ \begin{bmatrix} X \\ \alpha \end{bmatrix} \mid X \in \{e_1, e_2, \dots, e_n\}, \text{ and } \alpha \in \{h_{1,1}, h_{2,2}, \dots, h_{12,12}\} \right\}$ is the initial set of points where e_i is the i th column of the identity matrix and $h_{i,i}$ is the i th diagonal entry of H .

Table 1: Eigenvalues of H_{12} by Modified Newton Method

Eigenvalues	Eigenvalues of H	$\ (H - \alpha_k I)X_k\ _2$
1 st	1.7953720595620	4.5163365159057D-17
2 nd	0.38027524595504	9.5107769421299D-17
3 rd	4.4738548752D-02	9.3288777118150D-17
4 th	3.7223122378D-03	9.5107769421299D-17
5 th	2.3308908902D-04	6.5092594645258D-17
6 th	1.1163357483D-05	6.6374428417771D-17
7 th	4.0823761104D-07	1.9236667674542D-16
8 th	1.1228610666D-08	4.9553614188006D-17
9 th	2.2519644461D-10	6.0015952254039D-17
10 th	3.1113405079D-12	6.5125904614112D-17
11 th	2.6487505785D-14	1.0932505712948D-16
12 th	1.1161909467D-16	6.0015952254039D-17

If we have some clustering case on the target matrix, we can also overcome all eigenpairs [Figure 7].

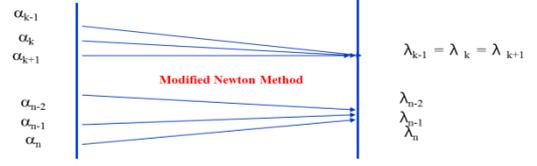


Fig 7. MNM Process.

To overcome the lost eigenpairs from MNM by the Gram-Schmidt orthogonalization process.

Suppose both $[x_1, \alpha_1]^T$ and $[x_2, \alpha_2]^T$ converge to $[U_1, \lambda_1]^T$.

(i) Choose a vector $x = x_1$ (or x_2).

(ii) Apply the Gram-Schmidt orthogonalization process.

$$z = x - \langle U_1, x \rangle U_1, \quad xx = z / \|z\|_2$$

(iii) Apply MNM with $[xx, \alpha_1]^T$ (or $[xx, \alpha_2]^T$).

$[xx, \alpha_1]^T$ (or $[xx, \alpha_2]^T$) will converge to $[U_2, \lambda_2]$.

It is shown the MNM in (10), (11) and this is algorithm as below:

Modified Newton's Algorithm

Set $\alpha_k^{(1)}$ and $x_k^{(1)}$.

$(\alpha_k^{(1)}, x_k^{(1)})$ is an initial eigenpairs where $\alpha_k^{(1)}$ and $x_k^{(1)}$, $k = 1, \dots, n$ are the diagonal entries of the given matrix and $\{e_1, e_2, \dots, e_n\}$.

For $k = 1, \dots, n$ do (in parallel)

For $j = 1, \dots$ do until convergence

(i) Solve for $y_k^{(j)}$, $(\alpha_k^{(j)} I - A)y_k^{(j)} = x_k^{(j)}$

(ii) Compute $\beta_k^{(j)} = x_k^{(j)T} \cdot y_k^{(j)}$

(iii) Compute $\check{\beta}_k^{(j)} = \|y_k^{(j)}\|_2$.

(iv) Compute $x_k^{(j+1)} = \frac{1}{\check{\beta}_k^{(j)}} y_k^{(j)}$

(v) Compute $\alpha_k^{(j+1)} = \alpha_k^{(j)} - \frac{1}{\beta_k^{(j)}}$

End

End

We apply the MNM (Modified Newton Method) to generate eigen-images at each step for image recognition problems.

In this case, we work to find the all eigenpairs of a covariance matrix which is generated by the characteristic vectors.

Step 4: Solving of the eigen problem of a Covariance Matrix C

We compute the eigenvalue λ_i and the corresponding eigenvectors μ_i of the covariance matrix C . It is based on our algorithm. The eigenvalue present an average face of the given image and it produce an eigen-face (Fig. 8).

$$\mu_l = \frac{1}{M} \sum_{n=1}^M \lambda_{lk} \Phi_k, \quad l=1, 2, \dots, M. \quad (13)$$

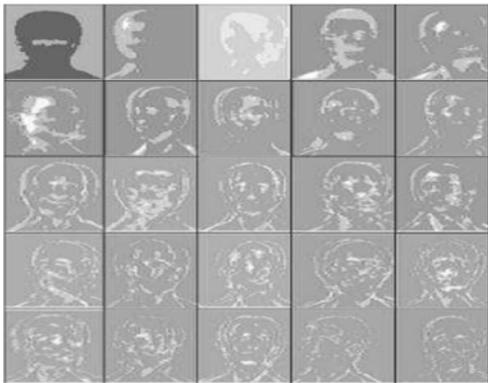


Fig. 8 Eigen-Faces

III. CONCLUSION

We can get each component ω_k to be projected by the eigenvalues λ_k after their new faces are input. It is shown how to compute each eigenface component as follows :

$$\omega_k = \mu_k^T (\Gamma - \Psi), \quad k=1, 2, \dots, M' \quad (14)$$

M' is the number that there are excluded the smallest eigenface corresponding the smallest eigenvalues. From the weight, we can get the component vector Ω of eigenface that are expressed input face images. If we get the set $\Omega = \{\omega_1, \omega_2, \dots, \omega_{M'}\}$, we can expect the errors as follows $\varepsilon_k = \left\| \Omega - \Omega_k \right\|_2$.

Step 5: A Reconstruction of the given image
 We compare the origin image and reconstructed image which is applied our algorithm (Figure 9).

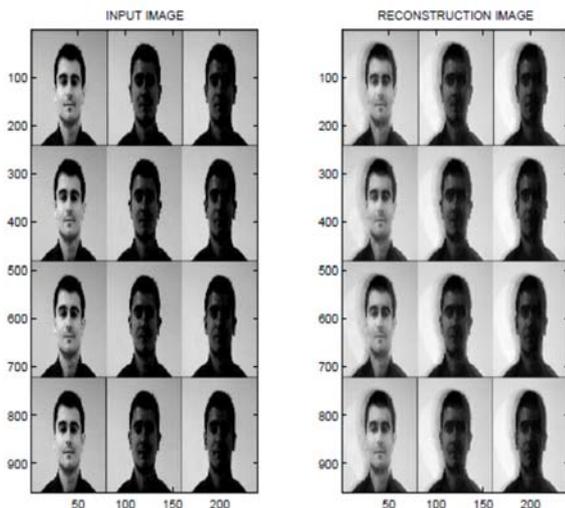


Fig. 9. An Input image and reconstruction of a Face with the algorithm

REFERENCES

- [1] A. Webb, *Statistical Pattern Recognition*, 2nd: John Wiley & Sons Inc., 2002.
- [2] J. M. Ortega, *Numerical Analysis, A Second Course: SIAM Series in Classical in Applied Mathematic*: Philadelphia, SIAM Publications, 1990.
- [3] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution Of Nonlinear Equations In Several Variables*: Academic Press, New York and London, 1970.
- [4] R. Baik, K. Datta and Y. Hong, "Computing for Eigenpairs on Globally Convergent Iterative for Hermitian Matrices," *Lecture Note in Computer Science*, vol. 3514, pp. 1- 8, 2005.
- [5] Karabi Datta and Yoopyo Hong and Ran Baik Lee, "Parameterized Newton's iteration for computing an Eigenpairs of a Real Symmetric Matrix in an Interval," *Computational Methods in Applied Mathematics*, Vol. 3, pp. 517-535, 2003.
- [6] S. Theodoridis and K. Koutrombas, *Pattern Recognition*: Academic Press, 1998.
- [7] [http:// gpstsc.upc.es/GTAV/ResearchAreas/UPCFace-Database/ GTAVFaceDatabase.htm](http://gpstsc.upc.es/GTAV/ResearchAreas/UPCFace-Database/GTAVFaceDatabase.htm), 2017.

Authors



Ran Baik has BS in mathematics from Sungkyunkwan University, MS in Applied Mathematics/Computer science from North Carolina State University, Doctorate in Numerical Analysis from Northern Illinois University, Doctoral course completion in Financial Engineering from Chonnan national university. Ran is currently a professor of dept. of Computer Engineering at Honam university, GwangJu, Korea. Major research Area is Numerical analysis (parallel algorithm) and Financial Engineering.

