Development of 3D Face Databases by Using Merging and Splitting Eigenspace Models

BENYAMIN KUSUMOPUTRO, YUDHI SATRIA
Computational Intelligence Research Laboratory
Faculty of Computer Science
University of Indonesia,
Depok Campus 16424
INDONESIA

Abstract– In recent years, progress has been made on the problem of face recognition, especially in head-on face images with controlled illumination and scale. Good results have been obtained for 2-D frontal images and many researchers are now trying to extent this high recognition capability of the system, to recognize more general view positions of images that cover the entire 3-D viewing sphere. It is argued that 3-D recognition can be accomplished using linear combinations of as few as four or five 2-D viewpoint images, however, there are some drawbacks, because the system should calculate quite many numbers of two-dimensional observed images at various visual points, and memorizing 3-D objects required large memory requirements. In this paper we present the development of 3D face images databases using the merging and splitting methods. The first method will merge two eigenspace models, where each eigenspace represent a set of *n*-dimensional observation into union of sets. The second method will split one model from another to represent the difference between the sets. Result of experiments show that the developed system has higher degree of similarity, even using 84% of eigenvectors.

Key words: 3D recognition system, eigenspace models, singular value decomposition, principal component analysis,

1 Introduction

Recently, the eigenspace representation of images has attracted a lot of researchers. In this transformed space, the individual features of the images could be designed to be uncorrelated, and its dimension will reduce significantly. For a given set of images, the eigenspace representation can be achieved by performing the Eigen Value Decomposition (EVD) (also called Principal Component Analysis) or Singular Value Decomposition (SVD). Both methods are well known techniques in image processing as *batch* computation, however, this batch computation has disadvantage on its high computational cost, due to all observations should be simultaneously computed at once. Another computational technique, the *incremental* method computes the eigenspace model by successively updating the earlier model when new observations are available and neccesary. This method does not need to compute all observations at once, thus, reduce storage requirements and opening the possibility of computing on its searching an image-data on a very large gallery of images. Other advantages of using incremental computation lie on its application when the gallery of images is dynamically changed.

Previous research in incremental computation of eigenspace models has only considered by adding each one new observation at a single time to the already developed eigenspace model [1], [2], [3], [4], [5]. These methods, however, ignore the fact that a change in incoming new data will also change its mean. More over, we know that when a few incremental updates were made, the inaccuracy was very small and acceptable for the great majority of applications; however, when thousands of updates were made frequently, the inaccuracies increases higher, and sometimes could not be tolerated.

Since eigenspace models have a wide variety of applications, Hall et all [6] then proposed the merging and splitting eigenspace model algorithms. Since the eigenspace model has been used frequently in the various classification problems, such as: recognition systems [7], motion sequence analysis [8], and temporal tracking of signals [4], the used of the merging and splitting of eigenspace models is then very useful. Using this model, building the eigenspace new model, due to its new coming data, could be accomplished without re-compute the previous eigenspace model. Hall et all also shows that the efficiency and accuracy of this method is better than the batch methods, on its performing to build a database of 2D face images.

We have developed a 3D face recognition system by using a cylindrical structure of hidden layer neural network (CSHL-NN) [9][10]. The input given is usually a number of two-dimensional images that are observed at the specific visual positions. It is very difficult; however, to realize a recognition system with high quality and high speed processing, because the system should calculate quite many numbers of two-dimensional observed images at various visual points. Another problem arises due to memorizing 3-D objects required large memory sizes, and when various directions of light beams are considered, even for the predetermined visual point, a large number of images should be calculated leading to larger memory requirements.

In order to increase its recognition rate of or developed system, especially, the ability to compute a huge gallery of face images, we then investigate the possibility of using merging and splitting eigenspaces techniques. In this paper, we would like to construct the eigenspace models of 3D face databases, and compare the accuracy of the images before and after performing the eigenspace transformation technique. We also investigated the performace of the eigenspace matrix transformer, through its value of cumulative percentage. This value is related to the degree of importance of the usage eigenvalue.

2 Merging and Splitting Eigenspace Models

Suppose an image of size *n x m* pixels is represented as a vector in an *n.m* dimensional space. In practice, however, this (*n.m)*-dimensional space is too large to process, so that a common way to resolve this problem is by using dimensionality reduction techniques. In its dimensionality reduction space, (*n.m)*-dimensional data should be firstly transformed into vector in the eigenspace model. For clarity, we define the nomenclature of an eigenspace model as Ω, with the mean vector ($\bar{\mathbf{x}}$), a (reduced) set of eigenvectors ($\mathbf{U}_{np}$), their eigenvalues ($\mathbf{\Lambda}_{pp}$), and number of observations (N): $\Omega = \left(\bar{\mathbf{x}}, \mathbf{U}_{np}, \mathbf{\Lambda}_{pp}, N\right)$.

Consider N observations, with each a column vector $\mathbf{x}^i \in \Re^n$, then we can compute the mean of the observations:

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{n}\mathbf{x}^i \qquad (1)$$

with their covariance:

$$\mathbf{C}_{nn} = \frac{1}{N}\sum(\mathbf{x}^i - \bar{\mathbf{x}})(\mathbf{x}^i - \bar{\mathbf{x}})^T \qquad (2)$$

Please note that $\mathbf{C}_{nn}$ is real and symmetric.

The eigenvalue decomposition (EVD) of $\mathbf{C}_{nn}$ can be calculated through:

$$\mathbf{C}_{nn} = \mathbf{U}_{nn}\mathbf{\Lambda}_{nn}\mathbf{U}_{nn}^T \qquad (3)$$

where the columns of $\mathbf{U}_{nn}$ are eigenvectors, and $\mathbf{\Lambda}_{nn}$ is a diagonal matrix of eigenvalues. The eigenvectors are orthonormal so that $\mathbf{U}_{nn}^T\mathbf{U}_{nn} = \mathbf{I}_{nn}$. The *i*th eigenvector $\mathbf{U}_{nn}^i$ and *i*th eigenvalue $\mathbf{\Lambda}_{nn}^{ii}$ are associated.

Typically, only $p \le \min(n, N)$ of the eigenvectors have significant eigenvalues and, hence, only *p* of the *n* eigenvectors need be retained. This happens when the observations are highly correlated so that the covariance matrix is, to a good approximation, rank degenerate. In this case, small eigenvalues are presumed to be negligible. Having chosen to discard certain eigenvectors and eigenvalues, we can recast Eq.3 using block from matrices and vectors. Without loss of generality, we can permute the eigenvectors and eigenvalues such that $\mathbf{U}_{np}$ and $\mathbf{\Lambda}_{pp}$ are those eigenvectors and eigenvalues that are kept, respectively, and $\mathbf{U}_{nd}$ and $\mathbf{\Lambda}_{dd}$ are those eigenvectors and eigenvalues that are discarded, with $d = n - p$. We may then rewrite Eq.3 as:

$$\mathbf{C}_{nn} = [\mathbf{U}_{np}\mathbf{U}_{nd}]\begin{bmatrix}\mathbf{\Lambda}_{nn} & \mathbf{0}_{pd}\\ \mathbf{0}_{dp} & \mathbf{\Lambda}_{dd}\end{bmatrix}[\mathbf{U}_{np}\mathbf{U}_{nd}]^T \approx \mathbf{U}_{np}\mathbf{\Lambda}_{pp}\mathbf{U}_{np}^T \qquad (4)$$

with error $\mathbf{U}_{nd}\mathbf{\Lambda}_{dd}\mathbf{U}_{nd}^T$, which is small if $\mathbf{\Lambda}_{dd} \approx \mathbf{0}_{dd}$

2.1 Construction of Eigenspace Models with SVD

In principle, computing an eigenspace model requires a construction of *(n x n)* matrix, where *n* is the dimension of each observation. In practice, however, the model can be computed by using an *(N x N)* matrix, where *N* is the number of observations. This is an advantage in its applications to the problem of image processing where, typically, $N << n$.

This technique can be done by considering the relationship between eigenvalue decomposition and singular value decomposition, which leads to a simple derivation for a low-dimensional batch method [11] on its computational of the eigenspace model.

Let $\mathbf{Y}_{nN}$ be the set of observations shifted to the mean so that $\mathbf{Y}^i = \mathbf{x}^i - \bar{\mathbf{x}}$. Then, an SVD of $\mathbf{Y}_{nN}$ is:

$\mathbf{Y}_{nN} = \mathbf{U}_{nn}\mathbf{\Sigma}_{nN}\mathbf{V}_{NN}^T$, where $\mathbf{U}_{nn}$ are the left singular vectors, which are identical to the eigenvectors previously given; $\mathbf{\Sigma}_{nN}$ is a matrix with singular values on its leading diagonal, with $\mathbf{\Lambda}_{nn} = \mathbf{\Sigma}_{nN}\mathbf{\Sigma}_{nN}^T / N$; and $\mathbf{V}_{NN}$ are right singular vectors. Both $\mathbf{U}_{nn}$ and $\mathbf{V}_{NN}$ are orthonormal matrices. This can now be used to compute eigenspace models in a low-dimensional way, as: $\mathbf{Y}_{nN}^T\mathbf{Y}_{nN} = \mathbf{V}_{NN}\mathbf{\Sigma}_{nN}^T\mathbf{\Sigma}_{nN}\mathbf{V}_{NN}^T = \mathbf{V}_{NN}\mathbf{S}_{NN}\mathbf{V}_{NN}^T$; which is an *(N x N)* eigenproblem. $\mathbf{S}_{NN}/N$ is the same as $\mathbf{\Lambda}_{nn}$ except for the presence of extra trailing zeros on the main diagonal of $\mathbf{\Lambda}_{nn}$. If we discard the small singular values, and their singular vectors, following the above, then remaining eigenvectors are $\mathbf{U}_{np} = \mathbf{Y}_{nN}\mathbf{V}_{Np}\mathbf{\Sigma}_{pp}^{-1}$.

2.2 Merging Process of Eigenspace Models

This section explained the process of merging the eigenspace models. We derive a solution to the following problem. Let $\mathbf{X}_{nN}$ and $\mathbf{Y}_{nM}$ be two sets of observations. Let their eigenspace models be $\Omega = (\bar{\mathbf{x}}, \mathbf{U}_{np}, \mathbf{\Lambda}_{pp}, N)$ and $\Psi = (\bar{\mathbf{y}}, \mathbf{V}_{nq}, \mathbf{\Delta}_{qq}, M)$, respectively. The problem is related to how to compute the eigenspace model $\Phi = (\bar{\mathbf{z}}, \mathbf{W}_{nr}, \mathbf{\Pi}_{rr}, K)$, for $\mathbf{Z}_{n(N+M)} = [\mathbf{X}_{nN}\mathbf{Y}_{nM}]$ using the already known Ω and Ψ.

Clearly, the total number of new observations is $K = N + M$, then the combined mean can be calculated through:

$$\bar{\mathbf{z}} = \frac{1}{K}(N\bar{\mathbf{x}} + M\bar{\mathbf{y}}) \qquad (5)$$

with the combined covariance matrix:

$$\begin{aligned}\mathbf{E}_{nn} &= \frac{1}{K}\left(\sum_{i=1}^{n}\mathbf{x}^i(\mathbf{x}^i)^T + \sum_{i=1}^{M}\mathbf{y}^i(\mathbf{y}^i)^T\right) - \mathbf{z}\mathbf{z}^T \\ &= \frac{N}{K}\mathbf{C}_{nn} + \frac{M}{K}\mathbf{D}_{nn} + \frac{NM}{K^2}(\bar{\mathbf{x}}-\bar{\mathbf{y}})(\bar{\mathbf{x}}-\bar{\mathbf{y}})^T\end{aligned} \qquad (6)$$

where the first two terms are the combine scaled versions of $\mathbf{C}_{nn}$ and $\mathbf{D}_{nn}$ for the covariance matrices of $\mathbf{X}_{nN}$ and $\mathbf{Y}_{nM}$, respectively, while the third term is related to the change of mean.

We then compute the *s* eigenvectors and eigenvalues that satisfy:

$$\mathbf{E}_{nn} = \mathbf{W}_{ns}\mathbf{\Pi}_{ss}\mathbf{W}_{ns}^T \qquad (7)$$

where some eigenvalues are subsequently discarded to give *r* nonnegligible eigenvectors and eigenvalues. The problem is then related to its dimension of *s*, which is bounded by

$$\max(p,q) \le s \le p + q + 1 \qquad (8)$$

Then we can now forming a new eigenproblem by writing such [6]:

$$\begin{aligned}\mathbf{E}_{nn} &= \frac{N}{K}\mathbf{C}_{nn} + \frac{M}{K}\mathbf{D}_{nn} + \frac{NM}{K^2}(\bar{\mathbf{x}}-\bar{\mathbf{y}})(\bar{\mathbf{x}}-\bar{\mathbf{y}})^T \\ &= [\mathbf{U}_{np}\mathbf{v}_{nt}]\,\mathbf{R}_{ss}\mathbf{\Pi}_{ss}\mathbf{R}_{ss}^T\,[\mathbf{U}_{np}\mathbf{v}_{nt}]^T\end{aligned} \qquad (9)$$

where:

$$\begin{aligned}\mathbf{v}_{nt} &= \text{Orthonormalize}([\mathbf{H}_{nq}\mathbf{h}]) \\ \mathbf{H}_{nq} &= \mathbf{V}_{nq} - \mathbf{U}_{np}\mathbf{U}_{np}^T\mathbf{V}_{nq} \\ \mathbf{h} &= (\bar{\mathbf{y}}-\bar{\mathbf{x}}) - \mathbf{U}_{np}\mathbf{U}_{np}^T(\bar{\mathbf{y}}-\bar{\mathbf{x}})\end{aligned} \qquad (10)$$

If we multiplied the left side of Eq.9 by $[\mathbf{U}_{np}\mathbf{v}_{nt}]^T$ and the right side by $[\mathbf{U}_{np}\mathbf{v}_{nt}]$; and by using the fact that $[\mathbf{U}_{np}\mathbf{v}_{nt}]^T$ is a left inverse of $[\mathbf{U}_{np}\mathbf{v}_{nt}]$, then we obtain:

$$\begin{aligned}&[\mathbf{U}_{np}\mathbf{v}_{nt}]^T\left(\frac{N}{K}\mathbf{C}_{nn} + \frac{M}{K}\mathbf{D}_{nn} + \frac{NM}{K^2}(\bar{\mathbf{x}}-\bar{\mathbf{y}})(\bar{\mathbf{x}}-\bar{\mathbf{y}})^T\right)[\mathbf{U}_{np}\mathbf{v}_{nt}] \\ &= \mathbf{R}_{ss}\mathbf{\Pi}_{ss}\mathbf{R}_{ss}^T\end{aligned} \qquad (11)$$

which is a new eigenproblem whose eigenvectors constitute the $\mathbf{R}_{ss}$. However, since $\mathbf{C}_{nn}$ and $\mathbf{D}_{nn}$ are unknown, we should change the Eq.11 to be Eq.12:

$$\begin{aligned}&\frac{N}{K}\begin{bmatrix}\mathbf{\Lambda}_{pp} & \mathbf{0}_{pt}\\ \mathbf{0}_{tp} & \mathbf{0}_{tt}\end{bmatrix} + \\ &\frac{M}{K}\begin{bmatrix}\mathbf{G}_{pq}\mathbf{\Delta}_{qq}\mathbf{G}_{pq}^T & \mathbf{G}_{pq}\mathbf{\Delta}_{qq}\mathbf{\Gamma}_{tq}^T\\ \mathbf{\Gamma}_{tq}\mathbf{\Delta}_{qq}\mathbf{G}_{pq}^T & \mathbf{\Gamma}_{tq}\mathbf{\Delta}_{qq}\mathbf{\Gamma}_{tq}^T\end{bmatrix} + \\ &\frac{NM}{K^2}\begin{bmatrix}\mathbf{g}_p\mathbf{g}_p^T & \mathbf{g}_p\boldsymbol{\gamma}_t^T\\ \boldsymbol{\gamma}_t\mathbf{g}_p^T & \boldsymbol{\gamma}_t\boldsymbol{\gamma}_t^T\end{bmatrix}(\bar{\mathbf{x}}-\bar{\mathbf{y}})(\bar{\mathbf{x}}-\bar{\mathbf{y}})^T = \mathbf{R}_{ss}\mathbf{\Pi}_{ss}\mathbf{R}_{ss}^T\end{aligned} \qquad (12)$$

where:

$$\begin{aligned}[\mathbf{U}_{np}\mathbf{v}_{nt}]\mathbf{C}_{nn}[\mathbf{U}_{np}\mathbf{v}_{nt}]^T &\approx \begin{bmatrix}\mathbf{\Lambda}_{pp} & \mathbf{0}_{pt}\\ \mathbf{0}_{tp} & \mathbf{0}_{tt}\end{bmatrix} \\ \mathbf{G}_{pq} &= \mathbf{U}_{np}^T\mathbf{V}_{nq} \\ \mathbf{\Gamma}_{tq} &= \mathbf{v}_{nt}^T\mathbf{V}_{nq} \\ \mathbf{g}_p &= \mathbf{U}_{np}^T(\bar{\mathbf{x}}-\bar{\mathbf{y}}) \\ \boldsymbol{\gamma}_t &= \mathbf{v}_{nt}^T(\bar{\mathbf{x}}-\bar{\mathbf{y}})\end{aligned} \qquad (13)$$

2.3 Splitting Process of Eigenspace Models

In this section we show how the process of splitting the combined eigenspace to be a two eigenspace models. Given a combined eigenspace model of $\Phi = (\bar{\mathbf{z}}, \mathbf{W}_{nr}, \mathbf{\Pi}_{rr}, K)$, in which we would like to remove $\Psi = (\bar{\mathbf{y}}, \mathbf{V}_{nq}, \mathbf{\Delta}_{qq}, M)$ to give a third model of eigenspace $\Omega = (\bar{\mathbf{x}}, \mathbf{U}_{np}, \mathbf{\Lambda}_{pp}, N)$

The splitting derivation process of combined eigenspace can be straightforwardly done inversely to that of the merging process. Let $N = K - M$, and the new mean is:

$$\bar{\mathbf{x}} = \frac{K}{N}\bar{\mathbf{z}} - \frac{M}{N}\bar{\mathbf{y}} \qquad (14)$$

As in the case of merging process that a new eigenvalues and eigenvectors are computed via a new eigenproblem, for splitting process, we have:

$$\frac{K}{N}\mathbf{\Pi}_{rr} - \frac{M}{N}\mathbf{G}_{rq}\mathbf{\Delta}_{qq}\mathbf{G}_{rq}^T - \frac{M}{K}\mathbf{g}_r\mathbf{g}_r^T = \mathbf{R}_{rr}\mathbf{\Lambda}_{rr}\mathbf{R}_{rr}^T \qquad (15)$$

where:

$$\begin{aligned}\mathbf{G}_{rq} &= \mathbf{W}_{nr}^T\mathbf{V}_{nq}\\ \mathbf{g}_r &= \mathbf{W}_{nr}^T(\bar{\mathbf{y}} - \bar{\mathbf{x}})\end{aligned} \qquad (16)$$

The eigenvalues that we would like to have are the p nonzero elements on the diagonal of $\mathbf{\Lambda}_{rr}$. Thus, we can permute $\mathbf{R}_{rr}$ and $\mathbf{\Lambda}_{rr}$, and rewrite the Eq.15, without loss of its generality, to be:

$$\begin{aligned}\mathbf{R}_{rr}\mathbf{\Lambda}_{rr}\mathbf{R}_{rr}^T &= [\mathbf{R}_{rp}\mathbf{R}_{tr}]\begin{bmatrix}\mathbf{\Lambda}_{rr} & 0_{pt}\\ 0_{tp} & 0_{tt}\end{bmatrix}[\mathbf{R}_{rp}\mathbf{R}_{tr}]^T\\ &= \mathbf{R}_{rp}\mathbf{\Lambda}_{pp}\mathbf{R}_{rp}^T\end{aligned} \qquad (17)$$

where: $p = r - q$.

Hence, we need only identify the eigenvectors in $\mathbf{R}_{rr}$ with nonzero eigenvalues and compute the $\mathbf{U}_{np}$ as:

$$\mathbf{U}_{np} = \mathbf{W}_{nr}\,\mathbf{R}_{rp} \qquad (18)$$

2.4 Data Reconstruction Process

Eigenspace model $\Omega = (\bar{\mathbf{x}}, \mathbf{U}_{np}, \mathbf{\Lambda}_{pp}, N)$ is then used to transform data $\mathbf{x}^i \in \Re^n$ such that

$$\mathbf{y}^i = \mathbf{U}_{np}^T(\mathbf{x}^i - \bar{\mathbf{x}}) \qquad (19)$$

and we can reconstruct equation (19) to get

$$\mathbf{x}^i = \mathbf{U}_{np}\mathbf{y}^i + \bar{\mathbf{x}} \qquad (20)$$

3 Experimental Set-Up and Its Results

The experimental procedure is conducted by using a 3-D face database that consists of 11 Indonesian persons. The images are taken under four different expressions such as, neutral, smile, angry and laugh expressions. The 2-D images are given from 3-D human face image by gradually changing visual points, which is successively varied from -90^0 to $+90^0$ with an interval of 15^0, i e -90^0, -75^0, -60^0, -45^0, -30^0, -15^0, 0^0, $+15^0$, $+30^0$, $+45^0$, $+60^0$, $+75^0$, $+90^0$. The total face images in the gallery that is utilized in this experiments are consists of 143 images, and part of them can be seen in Figure 2. The experiment are conducted using Matlab on a computer with standard configuration (Compaq, Pentium III, 128 Mb RAM).

3.1 Time for Merging Eigenspace Models

The merging eigenspace model is created by adding every 13 images into the already established eigenspace models. The number of eigenvectors that retain and used in every eigenspace model, is set to be a maximum. Computational result of merging time for every model of eigenspace is depicted in Figure 1. The total computational time is a combination of time-construction of eigenspace model and its merging-time, which can be categorises as the incremental time and the joint time.

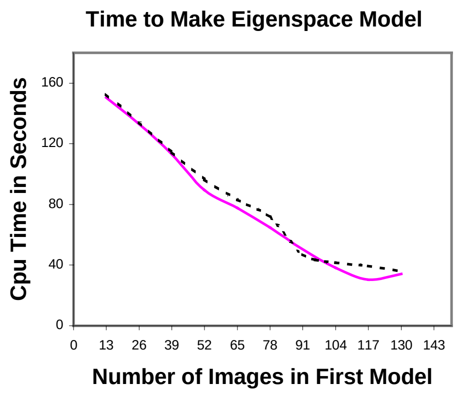

Figure 1. Computational time for complete eigenspace model of 143 images. (Solid line for the incremental time and dashed line for the joint time, respectively).

The *incremental time* is the time for computing and merging every one eigenspace model to the existing one, while the *joint time* is the time for computing both eigenmodels before merging and then to merge them. When measuring CPU time, we ran the same code several times and chose the fastest time to minimize the effect of other concurrently running process. As shown in Figure 1, *incremental time* and *joint time* are mostly the same.

3.2 Time for Splitting Eigenspace Models

Computational time for splitting eigenspace is done by removing every 13 images from a total 143 images. Result of experiments show that the average splitting time is approximately constant, range between *1.87* and *2.14* seconds, with a mean time around *1.933* seconds. These computational splitting times are much smaller than that of merging time, due to complicated calculation orthonormal basis.

3.3 Reconstruction Accuracy of 3D Face Images

In this type of experiments, we would like to elaborate the capability of the developed merging and splitting eigenface technique, to reconstruct the images with as small as possible error. Two types of gallery of images are developed; the first data set is the gallery images without background manipulation, while the second data set is for the gallery images with background manipulation.

The importance of every eigenvalue to its strength on giving the optimal matrix transformation for lower error rate, can be determined by computing the cummulative proportion of the eigen value using:

$$\alpha^k = \left(\sum_{i=1}^{k} \lambda_i\right) \Bigg/ \left(\sum_{j=1}^{p-1} \lambda_j\right) \qquad (21)$$

where α the cumulative percentage of the used eigenvalue, λ the eigenvalues that are nonnegative and arranged in decreasing order. The percentage of the used eigenvalue is *100%*, *98%*, *91%*, *84%* and *50 %*, respectively.

The visualization result for reconstructing 3D face images are shown in Figure 2, including with its relative mean errors that can be seen in Table 1. This result are just example of the overall data set, ranging from -90^0 to $+90^0$ with an interval of 15^0. The original images are depicted in the first line, while the second line is the reconstruction result by using 100% eigenvectors. The third line is the reconstruction result by using 98% eigenvectors that have largest eigenvalues, and the fourth line is the reconstruction result by using *91%* eigenvectors with the largest eigenvalues. The fifth line is the reconstruction result by using *84%* eigenvectors, while the sixth line is the reconstruction result by using *50%* of its eigenvectors.

The last column is the mean error for determining the degree of similarity between the original images (x) and the reconstruction result of images (y). This similarity degree is calculated through the equation of $\frac{\sum_{i=1}^{N} \frac{\|(y_i - x_i)\|}{\|x_i\|}}{N}$. While the overall similarity degree are shown in Table 1.

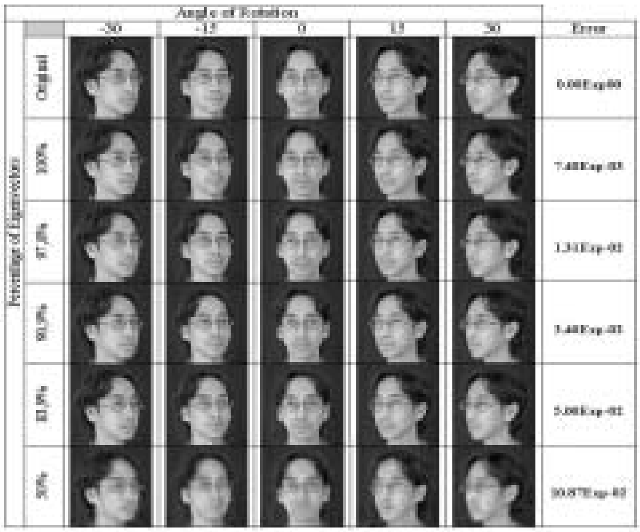

Figure 2. Reconstruction of images without background manipulation

It is clearly shown in this table, that the reconstruction of images is almost perfect for *100%* of used eigenface, with an error of only *0.0074*. The differences between original image values and reconstruction result image value is very small, however, as the used of eigenvector is decreasing, the mean error is increases. The relative mean errors are *0.0131* for *98%*, *0.034* for *91%* and *0.034* for *91%*, and *0.05* for *84%*, respectively. Please note that when using only *50%* of its eigenvectors, the relative mean error value is *0.1087*, which is not acceptable. It is visually shown that when using the eigenvector higher than *84%*, results are good and errors are acceptable.

Table 2 shows that the relative mean error when the number of the first eigenspace are different. When the first eigenspace is built from 13 images and added up to 143 images, the relative mean error still comparable with that of the first eigenspace is built from 130 images.

Table 1. Mean relative error of images without background manipulation

First images	100%	98%	91%	84%	50%
13	7.4000E-03	1.3100E-02	3.4000E-02	5.0000E-02	1.0870E-01
26	7.4000E-03	1.3100E-02	3.4000E-02	5.0000E-02	1.0870E-01
39	7.4000E-03	1.3100E-02	3.4000E-02	5.0000E-02	1.0870E-01
52	7.4000E-03	1.3100E-02	3.4000E-02	5.0000E-02	1.0870E-01
65	7.4000E-03	1.3100E-02	3.4000E-02	5.0000E-02	1.0870E-01
78	7.4000E-03	1.3100E-02	3.4000E-02	5.0000E-02	1.0870E-01
91	7.4000E-03	1.3100E-02	3.4000E-02	5.0000E-02	1.0870E-01
104	7.4000E-03	1.3100E-02	3.4000E-02	5.0000E-02	1.0870E-01
117	7.4000E-03	1.3100E-02	3.4000E-02	5.0000E-02	1.0870E-01
130	7.4000E-03	1.3100E-02	3.4000E-02	5.0000E-02	1.0870E-01

Table 2. Mean relative error of images without background manipulation from smaller to bigger of its first eigenspace model

	Jumlah Gambar									
eigenvalue	26	39	52	65	78	91	104	117	130	143
100%	0.0214	0.0144	0.0126	0.0084	0.0080	0.0068	0.0068	0.0066	0.0063	0.0062
98%	0.0316	0.0243	0.0169	0.0114	0.0133	0.0116	0.0107	0.0102	0.0114	0.0109
91%	0.0392	0.0377	0.0392	0.0295	0.0276	0.0246	0.0233	0.0230	0.0244	0.0240
84%	0.0624	0.0534	0.0511	0.0428	0.0401	0.0375	0.0359	0.0358	0.0361	0.0347
50%	0.1195	0.1088	0.1080	0.0974	0.0951	0.0915	0.0902	0.0867	0.0848	0.0808

The visualization result for reconstructing 3D face images with background manipulation are shown in Figure 3, and its relative mean errors are depicted in Table 3. As the same with that the images without background manipulation, results show that the merging and splitting eigenspace could be used for 3D faces databases.

As we can see in Table 3, the relative mean error still low enough when using *84%* of its eigenvectors or higher, i.e *0.033*. The background of the images is not considered to be an obstacle of the system, and the overall relative mean errors are not changing much by this difference.

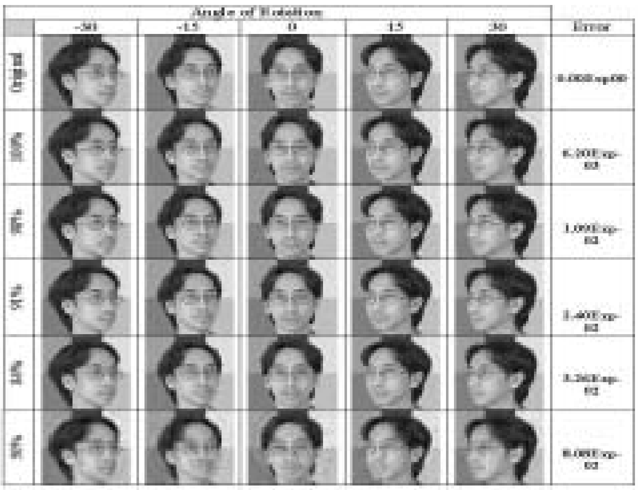

Figure 3. Reconstruction of images with background manipulation

As also has been shown for the images without background manipulation, increasing the number of images to be the first eigenfaces, will lowering the relative mean error slightly, and these values is still acceptable for 84% of the used eigenvectors or higher. These results are clearly shown in Table 4.

Table 3. Mean relative error of images with background manipulation

First images = 13					First images = 26				
100%	98%	91%	84%	50%	100%	98%	91%	84%	50%
6.2000E-03	1.0900E-02	2.4000E-02	3.4700E-02	8.8000E-02	6.2000E-03	1.0900E-02	2.4000E-02	3.4700E-02	8.8000E-02
First images = 39					**First images = 52**				
100%	98%	91%	84%	50%	100%	98%	91%	84%	50%
6.2000E-03	1.0900E-02	2.4000E-02	3.4700E-02	8.8000E-02	6.2000E-03	1.0900E-02	2.4000E-02	3.4700E-02	8.8000E-02
First images = 65					**First images = 78**				
100%	98%	91%	84%	50%	100%	98%	91%	84%	50%
6.2000E-03	1.0900E-02	2.4000E-02	3.4700E-02	8.8000E-02	6.2000E-03	1.0900E-02	2.4000E-02	3.4700E-02	8.8000E-02
First images = 91					**First images = 104**				
100%	98%	91%	84%	50%	100%	98%	91%	84%	50%
6.2000E-03	1.0900E-02	2.4000E-02	3.4700E-02	8.8000E-02	6.2000E-03	1.0900E-02	2.4000E-02	3.4700E-02	8.8000E-02
First images = 117					**First images = 130**				
100%	98%	91%	84%	50%	100%	98%	91%	84%	50%
6.2000E-03	1.0900E-02	2.4000E-02	3.4700E-02	8.8000E-02	6.2000E-03	1.0900E-02	2.4000E-02	3.4700E-02	8.8000E-02

Table 4. Mean relative error of images with background manipulation from smaller to bigger of its first eigenspace model

	Jumlah Gambar									
eigenvalue	26	39	52	65	78	91	104	117	130	143
100%	0.0238	0.0180	0.0166	0.0106	0.0100	0.0089	0.0081	0.0079	0.0076	0.0074
98%	0.0390	0.0278	0.0236	0.0161	0.0182	0.0168	0.0128	0.0143	0.0136	0.0131
91%	0.0518	0.0448	0.0489	0.0408	0.0391	0.0358	0.0326	0.0327	0.0341	0.0340
84%	0.0758	0.0648	0.0586	0.0584	0.0540	0.0519	0.0499	0.0519	0.0509	0.0500
50%	0.1487	0.1344	0.1321	0.1233	0.1204	0.1171	0.1155	0.1128	0.1117	0.1087

6 Conclusions

We have shown that merging and splitting eigenspace models for 3D face images is possible. Our experimental results show that there are no differences in CPU time between *incremental time* and *joint time.* The reconstruction results have shown that for all of the images used in these experiments, i.e images with or without background manipulation, the used of eigenvectors with 84% or higher will have an acceptable relative mean error.

Acknowledgement

The authors would like to express their gratitude to Ministry of Research and Technology of Indonesia for their support funding of this research through RUT X under contract number 14.06/SK/RUT/2003

References:

[1] J.R. Bunch, C.P. Nielsen, and D.C. Sorensen, ``Rank-One Modification of Symetric Eigenproblem", Numerische Mathematik, vol. 31, pp. 31-48, 1978.

[2] J.R. Bunch, C.P. Nielsen, ``Updating the

Singular Value Decomposition", Numerische Mathematik, vol. 31, pp. 111-129, 1978.

[3] S. Chandrasekaran, B.S. Manjunath, Y.F. Wang, J. Winkler, and H. Zhang, ``An Eigenspace Update Algorithm for Image Analysis", Graphical Models and Image Processing, vol. 59, no. 5, pp. 321-332, Sept. 1997.

[4] R.D. DeGroat and R. Roberts, ``Efficient, Numerically Stabilized Rank-One Eigenstructure Updating", IEEE Trans. Acoustic, Speech, and Signal Processing, vol. 38, no. 2, pp. 301-316, Feb. 1990.

[5] H. Murakami, and B.V.K.V. Kumar, ``Efficient Calculation of Primary Images from a Set of Images", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 4, no. 5, pp. 511-515, Sept. 1982.

[6] P. Hall, D. Marshall, and R. Martin, ``Merging and Splitting Eigenspace Models", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 9, pp. 1042-1049, Sept. 2000.

[7] B. Moghaddam and A. Pentland, ``Probabilistic Visual Learning for Object Representation", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 696-710, July 1997.

[8] S. Chaudhuri, S. Sharma, and S. Chartterjee, ``Recursive Estimation of Motion Parameters", Computer Vision and Image Understanding, vol. 64, no. 3, pp. 434-442, Nov. 1996.

[9] B. Kusumoputro and A. Sulita, Genetics algorithms in optimization of cylindrical-hidden layer neural network for 3-D object recognition system, *Neural Network World*, Vol.10, No. 4, 2000, pp. 631-639.

[10] B. Kusumoputro, 3-D Face Recognition System Using Cylindrical-Hidden Layer Neural Network and its optimization through genetic algorithms, *Proc. IASTED Artificial Intelligence*, 2002, pp. 118-123.

[11] G.H. Golub and C.F. Van Loan, ``Matrix Computation", Johns Hopkins, 1989.