# Face Image Compression using Generalized Hebbian Algorithm of Non-Parsed Image

Kyung-Hwa Lee, Seok-Bae Seo, Daijin Kim*, Dae-Seong Kang

School of Electrical , Electronic and Computer Engineering, Dong-A University.
Hadandong, Sahagu, Pusan, 604-714 KOREA

*Dept. of Computer Engineering, Pohang University of Sience and Technology,
San 31, Hyojadong,` Namgu, Pohang, Kyoungbuk, 790-784 KOREA

Tel:+82-51-200-7710, Fax: +82-200-7712
E-mail: dskang@daunet.donga.ac.kr

**Abstract**: This paper proposes an image compressing and template matching algorithm for face image using GHA (Generalized Hebbian Algorithm). GHA is a part of PCA (Principal Component Analysis), that has single-layer perceptrons and operates and self-organizing performance. We used this algorithm for feature extraction of face shape, and our simulations verify the high performance for the proposed method. The shape for face in the fact that the eigenvector of face image can be efficiently represented as a coefficient that can be acquired by a set of basis is to compress data of image. From the simulation results, the mean PSNR performance is 24.08[dB] at 0.047bpp, and reconstruction experiment shows that good reconstruction capacity for an image that not joins at leaning.

## 1. Intorduction

Face recognition by image processing is useful in several applications. In this paper, we are interested in representing the face image as a sum of coefficients by basis using GHA. The shape for face in the fact that the eigenvector of face image can be efficiently represented as a coefficient that can be acquired by a set of basis is to compress data of image.

There are many methods for face image compression and classification using GHA. Sanger is proposed GHA to solve an image coding problem [1]. The input image was coded using a linear feed forward network with a single layer of 8 neurons, each with 64 inputs.

Vic Brennan's method is based on parsing into non-overlapping 2X2 and blocks used the Haar transform for multiresolution [2]. Raoul Triller is proposed skin tumors characterization algorithm using GHA neural network [3]. Although can achieve high compressing rate and good performances, they only have local information of face image.

Instead of the parsed image approaches taken by these authors, we propose the non-parsed GHA that whole input image joined the learning of GHA neural network. The input image was coded using a linear feed forward network with forty neurons, each with the number of input image pixels (92X112). To train the GHA neural network, forty neurons were used for 92X112 full size image by a learning equation. Each of the forty neural makes displays the set of synaptic weight associated with a particular neuron of the network.

To code the image, each input image was multiplied by each of the forty basis, thereby generation of forty coefficients for image coding. And each coefficient was uniformly quantized with the number of bits approximately proportional to the logarithm of the variance of that coefficient over the image. To reconstruct the face image from the quantized coefficients, all the images were weighted by their quantized coefficients, and then added to reconstitute each block of the image.

Image database that is from the Olivetti Research Lab. The images are 92X112, 8-bit grayscale. Fig. 1. shows parts of input images used for training. Most images include over eighty percentage of face information

Fig. 1. Parts of input images for training

This work was supported by Korea Science and Engineering Foundation under the constract 1999-2-3-2-001-2.

## 2. Generalized Hebbian Algorithm

Generalized Hebbian learning is a basic unsupervised technique especially in feed-forward network. Oja has shown that simple Hebbian learning applied to single linear neuron extracts features describing best, in the mean-square error sense, the input data. In Hebbian rule, the learning term is proportional to the product of the input and the output of a neuron [4].

The computations involved in the GHA are simple; they may be summarized as follows:

1. Initialize the synaptic weights of the network, $w_{ji}$, to small random values at time $n$=1. Assign a small positive value to the learning rate parameter η.

2. For $n$=1, $j$=0, 1, ..., $m$-1, and $i$=0, 1, ..., $p$-1, compute

$$y_j(n)=\sum_{i=0}^{p-1} w_{ji}(n)\ x_i(n) \qquad (1)$$

$$\Delta w_{ij}(n)=\eta[A-B] \qquad (2)$$

$$A=y_j(n)\ x_i(n),$$

$$B=y_j(n)\sum_{k=0}^{j} w_{kj}(n)\ y_k(n)$$

where $x_i(n)$ is the $i$th component of the $p$-by-1 input vector $x(n)$ and $m$ is the desired number of principal components.

3. Increment $n$ by 1, go to step 2, and continue until the synaptic weights $w_{ji}$ reach their steady-state values.

For large $n$, the synaptic weight $w_{ji}$ of $j$ neuron converges to the $i$ th component of the eigenvector associated with the $j$ th eigenvalue of the correlation matrix of the input vector $x(n)$.

## 3. gha for face image

### 3.1 The input images

The Olivetti Research Lab has a face database containing 10 different pictures of 40 people. The images are 92X112, 8–bit grayscale. The pictures show variation in background lighting, scale, orientation, and facial expression. Individuals who used eyeglass were allowed to pose both with and without eyeglasses. Some individuals looked very similar to others.

### 3.2 GHA for face image compression

Instead of the parsed image approaches taken by most of authors, we propose the non-parsed GHA that whole input image joined the learning of GHA neural network. The input image was coded using a linear feed forward network with forty neurons, each with the number of input image pixels. The proposed image compression algorithm explained by the following procedure of five steps.

Step 1. Set compressing ratio and basis number ($m$). Compression ratio was represented by the bpp (bit per pixel) and basis number was less than the number of input images.

Step 2. Initialize basis($w_{ji}$) with small number. They have random values between –0.2 and 0.2 by random number generator.

Step 3. Learning the basis by equation (1) and (2).

Step 4. Compute GHA coefficients using the learned basis by equation (3) and (4), and approximate each coefficient by assigned bits.

$$C=\{\ c_0,\ c_1,\ \ldots,\ c_{m-1}\ \} \qquad (3)$$

$$c_j=\sum_{i=0}^{p-1} w_{ji}(n)\ x_i(n) \qquad (4)$$

Step 5. Compress face images using GHA coefficients($C$) by Step 4 and basis($w_{ji}$) by Step 3.

### 3.3 Template matching for face image

For Template matching for face image, we only used GHA coefficients. As basis have the information of face shape and ordered by the eigenvalue, the largest eigenvalue occupied high proportion for the input images by the equation (4). Fig. 2 shows the block diagram of proposed template matching algorithm. We used the Euclidian distance to compute minima distances between query image and matched image in the input images.

## 4. simulation results

### 4.1 GHA for face image compression

Fig. 3 shows parts of learned basis by GHA. Each of basis displays the set of synaptic weights associated with a particular neuron of network. Specifically, excitatory synapses are shown white, whereas inhibitory synapses are shown gray. In our experiment, each basis represents the columns of the 92X112 synaptic weight matrix $W^T$ after the generalized Hebbian algorithm has converged. To code the image, each input image was multiplied by each of the forty masks, thereby generation forty coefficients for image coding. And each coefficient was uniformly quantized with a number of bits approximately proportional to the logarithm of the variance of that coefficient over the image. To reconstruct the face image from the quantized coefficients, all the masks were weighted by their quantized coefficients, and then added to reconstitute each block of the image. The Olivetti Research Lab. image database has been used to experiment several face recognition algorithms. The images are 92X112, 8-bit grayscale.

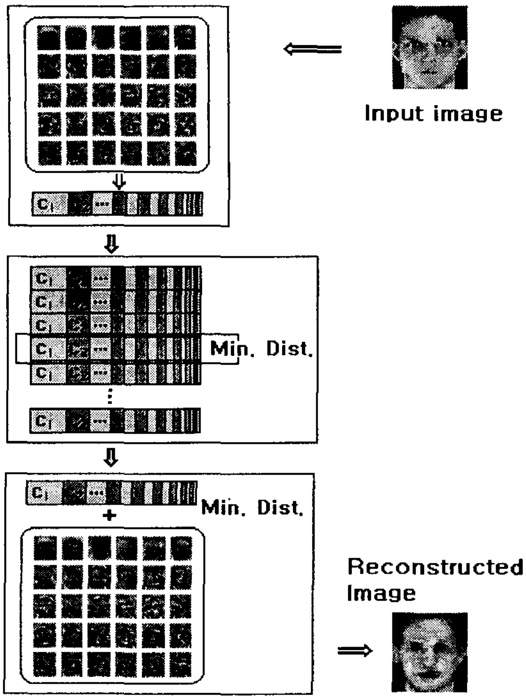


Fig. 2. The block diagram of proposed algorithm

Fig. 3. Learned basis images by GHA.

Each coefficient was uniformly quantized with a the number of bits approximately proportional to the logarithm of the variance of that coefficient over the image. Thus, the first basis was assigned 8 bit, from the second basis to twenty assigned 4 bit, from the twenty-one to thirty 3 bit, and the remaining ten basis 2 bit each. Base on this representation, a total of 136 bits were needed to code each 92X112 input images, resulting in a data rate of 0.047 bpp (bit per pixel). The result of mean PSNR value for input image is 24.08[dB], maximum is 26.71[dB], and minimum is 21.54[dB]. Fig 4 shows the reconstructed images by proposed algorithm.

To test performance of the proposed algorithm, an input image was used to code an image, that had not joined the learning by the network. Fig 5 shows the result images with similar appearance to those of the input images. The PSNR performance for this image is 20.72[dB].

Fig. 4. Reconstruction results for Fig2.

### 4.2 Template matching for face image

The function of template matching applied GHA is expressed to equation (5) that has the better similarity at smaller value. Where the coefficients of input images are given $C_{query}$, and of truncated $C_{query}$ are given $C_{aprox}$ that expressed equation (6). $T_{[b1,b2]}(n)$ is a function for truncations of float number n that use each assigned bit $b_1$ and $b_2$. If n has a sign of minus, the bit number of n is reduced one bit for represent the sign of minus. For examples, $T_{[0,3]}(0.9)=0.875$, $T_{[0,3]}(-0.9)=-0.750$, and $T_{[3,0]}(9)=7$.

$$\mathrm{Distc} = f\,(C_{query,}, C_{aprox}) \qquad (5)$$

$$= \sum_{i=0}^{p-1} \left( c_{query}, i,\ c_{aprox} \right)^2$$

$$C_{aprox} = T_{[0,3]}(C) \qquad (6)$$

Fig 6 shows a query image and result images by resemblance image searching using GHA coefficients. Results images have similar shapes for the input image.

Fig. 5. The Image that not join leaning of GHA and It's result image. (a) Input image, (b) reconstruction image.

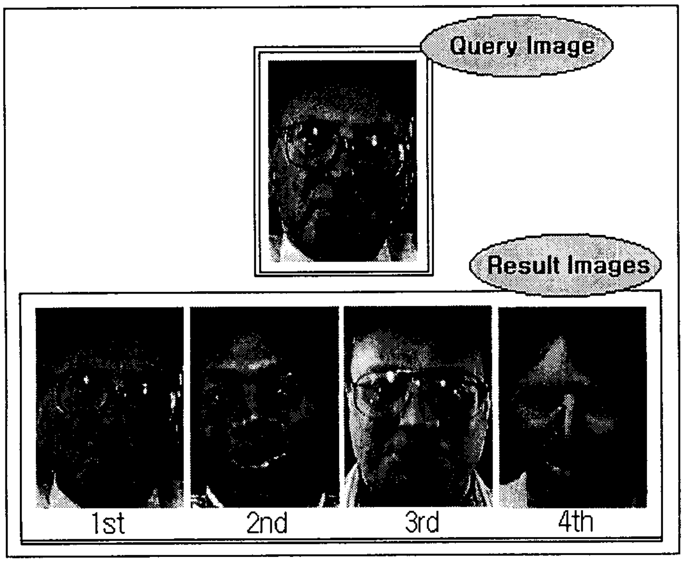


Fig. 6. Query image and resemblance images

## 5. Conclusions

In this paper, we have proposed a new algorithm for compression and template matching of face image. An advantage of the algorithm is based on shapes of face for image representation. The mean PSNR performance is 24.08[dB] at 0.047bpp, and reconstruction experiment shows that good reconstruction capacity for an image that not joins at leaning. In the reconstruction experiments of template matching, it can search images that resemble closely face shapes of query images. To problem with the proposed method is that the histograms of learning image are quite sensitive to the basis of GHA. To improve the robustness of the algorithm, we have also attempted to use histogram enhancement algorithm and face image preprocessing.

## References

[1] Haykin and Simon, "Neural Networks," Macmillan Publishing Company, pp.374-382, 1994.

[2] Vic Brennan and Jose Principe, "Face Classification using a Multiresolution Principal Component Analysis," Proceedings of the 1998 IEEE Workshop Signal Processing Society Workshop, pp.506-515, 1998.

[3] Raoul Triller and Bernard Fertil., "Optimal unsupervised learning in a single-layer linear feedforward neural network." Proceedings of the Applications of Artificial Neural Networks in Image Processing IV, pp.35-44 , 1999.

[4] Oja. E., "A simplified neuron model as a principal component analyzer," Journal of Mathematical Biology 15, pp.267-273, 1982.