

A Codebook Design Method for Robust VQ-Based Face Recognition Algorithm

Qiu Chen¹, Koji Kotani², Feifei Lee¹, Tadahiro Ohmi¹

¹New Industry Creation Hatchery Center, Tohoku University; ²Department of Electronics, Graduate School of Engineering, Tohoku University, Japan..

Email: qiu@fff.niche.tohoku.ac.jp

Received September 5th, 2009; revised November 2nd, 2009; accepted November 6th, 2009.

ABSTRACT

In this paper, we present a theoretical codebook design method for VQ-based fast face recognition algorithm to improve recognition accuracy. Based on the systematic analysis and classification of code patterns, firstly we theoretically create a systematically organized codebook. Combined with another codebook created by Kohonen's Self-Organizing Maps (SOM) method, an optimized codebook consisted of 2×2 codevectors for facial images is generated. Experimental results show face recognition using such a codebook is more efficient than the codebook consisted of 4×4 codevector used in conventional algorithm. The highest average recognition rate of 98.6% is obtained for 40 persons' 400 images of publicly available face database of AT&T Laboratories Cambridge containing variations in lighting, posing, and expressions. A table look-up (TLU) method is also proposed for the speed up of the recognition processing. By applying this method in the quantization step, the total recognition processing time achieves only 28 msec, enabling real-time face recognition.

Keywords: Face Recognition, Vector Quantization (VQ), Codebook Design, Code Classification, Histogram Method

1. Introduction

After September 11th, security systems utilizing personal biometric features, such as, face, voice, fingerprint, iris pattern, etc. are attracting a lot of attention. Among them, face recognition have become the subject of increased interest [1], which seems to be the most natural and effective method to identify a person since it is the same as the way human does and there is no need to use special equipments. In face recognition, personal facial feature extraction is the key to creating more robust systems.

A lot of algorithms have been proposed for solving face recognition problem. Based on the use of the Karhunen-Loeve transform, PCA [2] is used to represent a face in terms of an optimal coordinate system which contains the most significant eigenfaces and the mean square error is minimal. However, it is highly complicated and computational-power hungry, making it difficult to implement them into real-time face recognition applications. Feature-based approach [3,4] uses the relationship between facial features, such as the locations of eye, mouth and nose. It can implement very fast, but recognition rate usually depends on the location accuracy of facial features, so it can not give a satisfied recognition result. There are many other algorithms have been

used for face recognition. Such as Local Feature Analysis (LFA) [5], neural network [6], local autocorrelations and multi-scale integration technique [7], and other techniques [8–14] have been proposed.

Kotani *et al.* [15] have proposed a novel information-processing algorithm called Vector Quantization (VQ) codebook space information processing which differs from the traditional ways of processing algorithm. Based on this algorithm, we have developed a very simple yet highly reliable face recognition method called *VQ histogram method* by using a systematically organized Codebook for 4×4 blocks with 33 codevectors having monotonic intensity variation without DC component.

VQ algorithm [16] is well known in the field of image coding (compression). Input image is first divided into small blocks, which are taken as input vectors in VQ operation. Each input vector is then matched with codevectors in a codebook by calculating distances between them. The codevector having the maximum similarity to the input vector is selected by searching the minimum distance and the index number of the selected codevector is output.

This index number information was paid attention to. It was found that a codevector histogram, which is obtained by counting the matching frequency of individual

codevector, contains very effective facial feature information. By utilizing this technique, a novel face recognition algorithm called VQ histogram method has been developed.

A codebook is very important since it directly affects the quality of VQ processing. In [15], a special codebook was used, which is systematically organized for 4×4 blocks with 33 codevectors having monotonic intensity variation without DC component.

In this paper, a theoretical codebook design method is proposed. At first, a systematically organized codebook is created based on the distribution of code patterns abstracted from facial images [21], and then another codebook with the same size is created using Kohonen's Self-Organizing Maps (SOM) [20]. Combining the two codebooks obtained above, final optimized codebook consisted of 2×2 codevectors for facial images will be generated [22,23]. It can represent the features of the facial images more adequately. Furthermore, a table look-up (TLU) method is also proposed for the speed up of the recognition processing.

This paper is organized as follows. First, VQ histogram method will be introduced in detail in Section 2. Proposed codebook design method combining classification of code patterns and Kohonen's Self-Organizing Maps (SOM) will be described in Section 3. Experimental results compared with the algorithms employing original codebook or SOM codebook separately will be discussed in Section 4. Finally, we make a conclusion in Section 5.

2. Vector Quantization Histogram Method

In this section, we will describe the face recognition algorithm using Vector Quantization (VQ) histogram method [15]. Figure 1 shows face recognition process steps. First, low-pass filtering is carried out using simple 2-D moving average filter. This low-pass filtering is essential for reducing high-frequency noise and extracting most effective low frequency component for recognition. Block segmentation step, in which facial image is divided into small image blocks (for example, 2×2) with overlap, namely, by sliding dividing-partition one pixel by one pixel, is the following. Next, minimum intensity in the individual block is searched, and found minimum intensity is subtracted from each pixel in the block. Only the intensity variation in the block is extracted by this process. This is very effective for minimizing the effect of overall brightness variations. Vector quantization is then applied to intensity-variation blocks (vectors) by using a codebook which prepared in advance. The most similar (matched) codevector to the input block is selected.

After performing VQ for all blocks divided from a facial image, matched frequencies for each codevector are counted and histogram is generated. This histogram be-

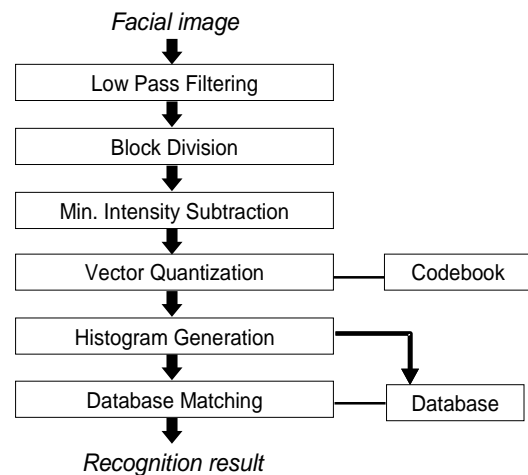


Figure 1. Face recognition process steps

comes the feature vector of human face. In the registration procedure, this histogram is saved in a database as personal identification information. In the recognition procedure, the histogram made from an input facial image is compared with registered individual histograms and the best match is output as a recognition result. Manhattan distance between histograms is utilized as a matching measure.

Codebook which consists of typical feature patterns for representing the features of face image is very important. In [15], 32 codevectors of 4×4 codebook are created by changing the direction (8 different directions) and the range of intensity variation (Step values are 2, 6, 10, and 20). By adding one codevector having no intensity variation, complete codebook is organized.

In the next section, we will propose a novel codebook design method for 2×2 codebook, which can represent the features of the facial image more adequately.

3. Codebook Design

Because of the characteristic of face image, the codevectors of 4×4 codebook are all low-frequency patterns. Although the average face recognition rate of 95.6% has been obtained using such a codebook, it is still difficult to say this codebook is the most suitable because the number of categories for the 4×4 block patterns is too large to be classified by only 33 categories, and it has not been proved in theory that these 33 codevectors most adequately represent the face patterns. But in the case of 2×2 code patterns, the condition is considerably different. Nakayama *et al.* [17] have developed a complete classification method for 2×2 codebook design for image compression. By the similar consideration, we classify and analyze the code patterns in the face images, and then theoretically create a new codebook of 2×2 code patterns for face recognition algorithm.

3.1 Previous Work

Nakayama *et al.* [17] proposed a complete classification method for 2×2 codebook design in image compression. Figure 2 shows all categories for the 2×2 image block patterns without considering the location of pixels. In a 2×2 block, pixel intensities are marked by alphabet 'a', 'b', 'c', 'd', and $a > b > c > d$ is prescribed. In [17], it was found that the number of typical patterns for all 2×2 image block is only 11. The number of varieties in pixel arrangement of each 2×2 typical pattern is also shown in Figure 2. That means the total number of image patterns for 2×2 pixel blocks is theoretically only 75.

By the similar consideration, we classified and analyzed the code patterns in the face images [21]. We found that in all filter size, the numbers of code patterns belong to categories 7, 10, and 11 are very few. It means such code patterns are almost not used in face images. Based on this result, we created a new codebook for 2×2 code patterns, and the rules of codebook creation are as follows.

1) Create very small intensity-variation (intensity difference among the block is only 1 or 2) code patterns having monotonic intensity variation, the number of code pattern is 16 by changing the direction of intensity variation.

2) Create code patterns of category No.2, 3, 4, 5, 6, 8 and 9, and intensity differences among the blocks are set to be 3, 6, and 10.

3) Do not make code patterns of category No. 7, 10, and 11.

4) Add one code pattern having no intensity variation.

Thus, complete codebook is systematically organized with 169 code patterns.

By using publicly available face database of AT&T Laboratories Cambridge [19], highest average recognition rate of 97.4% is obtained. Compared to the results of 4×4 codebook which the highest average recognition rate is 95.6%, recognition rate increases by about 2%.

But in [22], it was found that such distribution of codevectors appears non-uniform and concentrated only in some of the regions. According to *Maximum Entropy Principle* (MEP), the maximum entropy distribution

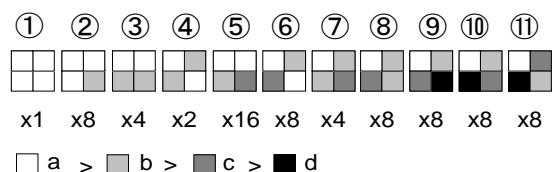


Figure 2. Categories of 2×2 code patterns. Pixel intensities are marked by alphabet 'a', 'b', 'c', 'd', and $a > b > c > d$ is prescribed. The number of typical patterns for all 2×2 image block is only 11, thus total number of image patterns for 2×2 pixel blocks is theoretically only 75

will be achieved when the value of a random variable (counts) equals the average. Such a non-uniform distribution can not satisfy the MEP, so the recognition performance can not be expected be best because the average information content will not be maximum.

As a solution, Chen *et al.* [22] optimized the codebook by sorting the frequencies of all individual codevectors abstracted from facial images and excluding the codevectors in the codebook with low frequency. This method improved recognition performance and highest average recognition rate of 98.2% is obtained by using the same face database of AT&T Laboratories Cambridge [19].

3.2 Proposed Codebook Design Method

The essence of the method in [22] is to abstract the code patterns which are most frequently used in facial images. The consideration is correct, but it ignored the differences between different persons. The frequencies of some code patterns used in facial images may vary greatly for different persons and the values appear small. The code patterns selected to be used in codebook should not only present the most common features of faces, but also discriminate the difference of persons. The latter characteristic is very important in recognition task and can be evaluated by *mean square error* (MSE) of code patterns. Figure 3(a) shows the average histogram of 40 facial images in the database of AT&T Laboratories Cambridge [19] by using the 2×2 codebook generated according to the method proposed in [22]. In this case, the codebook size is set to be 80. The distribution is sorted in order of frequency. But the order of respective MSE values is out of accord as shown in Figure 3(b). The code patterns with high frequencies but low MSE values are useful to generate feature vectors but poor benefit recognition result. So we should consider both factors of frequencies and MSE values.

As a solution, our strategy is as follows.

Step 1: Create a systematically organized codebook by applying an improved method based on the code classification including frequencies and MSE values.

Step 2: Create another codebook with the same size using Kohonen's Self-Organizing Maps (SOM) [20].

Step 3: Combine the two codebooks obtained above to generate the final optimized codebook consisted of 2×2 codevectors.

3.3 Data Set for Codebook Design

For covering the variations of the photo-taking conditions, two different face databases which are publicly available FERET database [18] and face database of AT&T Laboratories Cambridge [19], are utilized to analyze the code patterns of facial images. 40 facial images from different persons are selected from each database, and the face regions are abstracted with the sizes of 146×200 and 92×112 , respectively. The typical examples of facial images are shown in Figure 4.

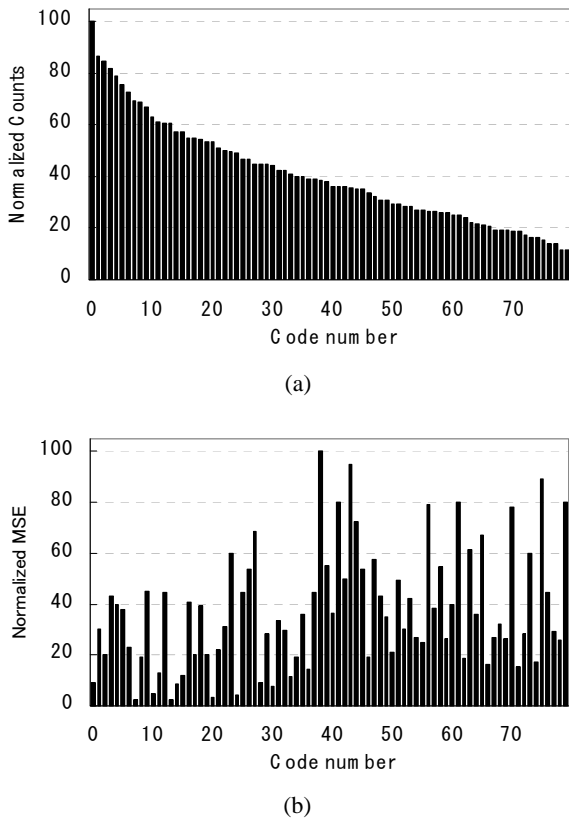


Figure 3. (a) Average histogram of 40 facial images using codebook of size 80. (b) MSE of code patterns. The distribution is sorted in order of frequency in (a). But the order of respective MSE values is out of accord as shown in (b)

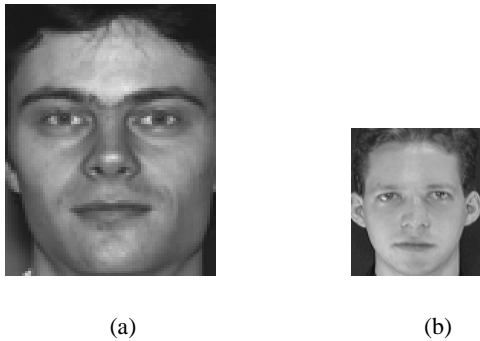


Figure 4. Data set for analysis of code patterns: (a) FERET database, (b) AT&T database

3.4 Codebook Generated by Code Classification

In the rules No. 2 of codebook creation in Subsection 3.1, the intensity variation steps are set to be 3, 6, and 10. From the distribution of the average histogram, we can see the steps for creating code patterns are not suitable obviously. We modify the rules and propose our improved codebook design method.

Figure 5 shows the processing steps of the codebook generation. At first, we change intensity differences among the blocks to be from 1 to 10, and implement the rules No.1–4 in Subsection 3.1 to create an initial large codebook of size 517. Thus the code patterns can almost cover all intensity variation. Utilizing this initial codebook, VQ processing is performed for all intensity variation blocks divided from the facial image in dataset described above, matched frequencies for each codevector are counted and histogram of each facial image is generated. Then average histogram and the normalized frequencies (\bar{f}_i) are calculated, where M is the number of facial images. MSE (e_i) (using RMSE, the square root of MSE) of individual codevectors is calculated by Formula 1 and then the scores (s_i) are computed by the weighted average between the normalized frequencies (\bar{f}_i) and MSE (e_i) as shown in Formula (2).

$$e_i = \sqrt{\frac{1}{M} \sum_{j=1}^M (\bar{f}_i - f_i(j))^2} \tag{1}$$

$$s_i = \frac{k_1 \bar{f}_i + k_2 e_i}{\sum_{j=1}^2 k_j} \tag{2}$$

where k_i ($i=1, 2$) is a weighting coefficient of respective component. The values of k_1, k_2 are 1, 1 respectively for all images used in our experiments, which determined by actual experiments.

Next, the scores (s_i) of individual codevectors are sorted, and the codevectors with high scores will be extracted. In this way, a systematically organized codebook is generated.

3.5 Codebook Generated by Kohonen’s SOM

As a neural unsupervised learning algorithm, Kohonen’s Self-Organizing Maps (SOM) [20] is one of the

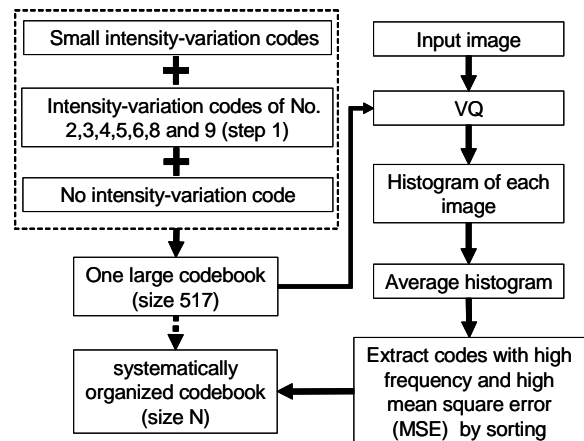


Figure 5. Processing steps of codebook generation based on code classification

standard algorithms used for codebook design in image coding. In this paper, SOM algorithm will also be employed to generate a SOM codebook with the same size as in Subsection 3.4.

Step 1: Transform the facial images in dataset to intensity variation vectors, and combine all data together into one training set.

Step 2: Specify the size of the codebook to N and initialize the codewectors by using continuous intensity variation vectors.

Step 3: Select a new training vector from the training set.

Step 4: Find the best-matching codewector closest to the training vector.

Step 5: Move the best-matching and its neighborhood codewectors towards the training vector.

Step 6: Repeat from Step 3 until the map converges.

3.6 Generation of Optimized Codebook

After the two codebooks described above are generated, they will be combined into one codebook of size $2N$. Overlapped codewectors in codebook will be excluded. Next, like the processing in Subsection 3.4, VQ processing is also performed for all intensity variation blocks divided from the facial image in dataset, and average histogram of all images is calculated. The scores of all individual codewectors are calculated and sorted, and the codewectors in this codebook with low scores will be excluded. Thus, the size of codebook will be decreased from $2N$ to N , and the final optimized codebook consisted of 2×2 codewectors is generated.

4. Experiments and Discussions

4.1 Database

Face database of AT&T Laboratories Cambridge [19] is used for recognition experiments. In the database, 10 facial images for each of 40 persons (totally 400 images) with variations in face angles, face sizes, facial expressions, and lighting conditions are included. Each image has a resolution of 92×112 . Five images were selected from each person's 10 images as probe images and remaining five images are registered as album images. Recognition experiment is carried out for 252 ($_{10}C_5$) probe-album combinations by rotation method. The algorithm is programmed by ANSI C and run on PC (Pentium(R)D processor 840 3.2GHz).

4.2 Experimental Results and Discussions

Firstly, we discuss the codebook size N . It is necessary to choose a suitable size of codebook. As the codebook size is too large, number of codewectors increases, the resolution of the histogram may become so sensitive that noise-corrupted codewectors may significantly distort the histogram. On the contrary, if the number of codewectors is too small, the histogram can not sufficiently discriminate between different faces.

Figure 6 shows the comparison of the recognition

results using codebooks with different sizes from 30 to 200. The highest average recognition rates obtained in each codebook are shown here. The best performance is obtained at codebook size of 80 which is the same as in [22]. Maximum of the average rate 98.6% is achieved, which is 0.4% higher than that in [22].

We also compare the proposed algorithm with conventional algorithms. Figure 7 shows the recognition results. Recognition success rates are shown as a function of filter size which changes from none filtering to filter size of 23×23 . The curve with rhombus marks stand for the average results in 252 ($_{10}C_5$) probe-album combinations using the 2×2 codebook created by proposed design method. The codebook size of 80 is used here. The curve with triangle marks and circular marks refer to the results reported in [21,22] which use original 2×2 codebook. The curve with open square marks refers to the results of 4×4 codebook [15]. In the curve with rhombus marks, recognition rate first increases with increase in filter size, and then, saturated or gradually decreases. The highest average recognition rate of 98.6% is obtained at the filter size of 13×13 while that of 98.2%, 97.4% and 95.6% using codebooks in [15,21,22], respectively. It can be said

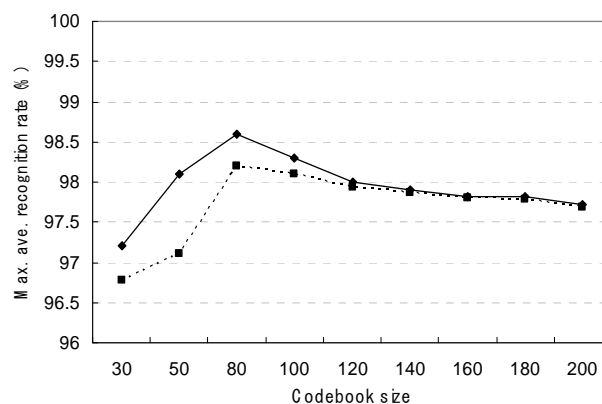


Figure 6. Comparison of recognition results in different codebook size

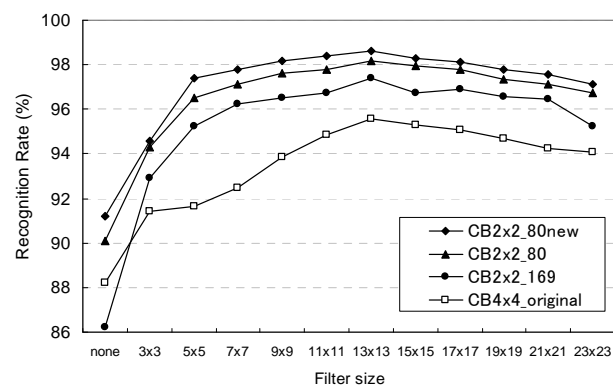


Figure 7. Comparison of the recognition results

that the 2×2 codebook created by proposed method represent the features of the facial images more adequately than conventional codebooks.

4.3 Processing Time

Processing time for single image in the face database of AT&T Laboratories Cambridge [19] is about 57 msec using a codebook of size 80, which is composed of 15 msec for pretreatment including filtering, block division, and minimum intensity subtraction, 30 msec for VQ processing, and 12 msec for database matching.

Furthermore, because a 2×2 codevector can be represented by an array of 4 dimensions, by utilizing the table look-up (TLU) method in the VQ processing step, the VQ processing time can be shorten to be about 1 msec, and the total running time will be 28 msec. It means our fast recognition algorithm achieves real-time face recognition.

5. Conclusions

In this paper, a theoretical codebook design method for robust VQ-based face recognition algorithm is proposed. Combining a systematically organized codebook based on the classification of code patterns and another codebook created by Kohonen's Self-Organizing Maps (SOM), an optimized codebook consisted of 2×2 codevectors for facial images is generated. Utilizing such a codebook of size 80, the highest average recognition rate of 98.6% is obtained for 40 persons' 400 images of the database of AT&T Laboratories Cambridge.

REFERENCES

- [1] K. W. Bowyer, "Face recognition technology and the security versus privacy tradeoff," *IEEE Technology and Society*, pp. 9–20, Spring 2004.
- [2] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, pp. 71–86, March 1991.
- [3] L. Wiskott, J. M. Fellous, N. Kruger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 775–779, July 1997.
- [4] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 15, No. 10, pp. 1042–1052, October 1993.
- [5] P. S. Penev and J. J. Atick, "Local feature analysis: A general statistical theory for object representation," *Network: Computation in Neural Systems*, Vol. 7, No. 3, pp. 477–500, 1996.
- [6] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," *Proceedings IEEE*, Vol. 83, No. 5, pp. 705–740, May 1995.
- [7] S. Z. Li and A. K. Jain, "Handbook of face recognition," Springer, New York, 2005.
- [8] F. Goudail, E. Lange, T. Iwamoto, K. Kyuma, and N. Otsu, "Face recognition system using local autocorrelations and multi-scale integration," *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 18, No. 10, pp. 1024–1028, 1996.
- [9] K. M. Lam and H. Yan, "An analytic-to-holistic approach for face recognition based on a single frontal view," *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 20, No. 7, pp. 673–686, 1998.
- [10] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 696–710, 1997.
- [11] W. Zhao, "Discriminant component analysis for face recognition," *Proceedings ICPR'00, Track 2*, pp. 822–825, 2000.
- [12] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, "Face recognition by independent component analysis," *IEEE Transactions on Neural Networks*, Vol. 13, No. 6, pp. 1450–1464, 2002.
- [13] S. G. Karungaru, M. Fukumi, and N. Akamatsu, "Face recognition in color images using neural networks and genetic algorithms," *International Journal of Computational Intelligence and Applications*, Vol. 5, No. 1, pp. 55–67, 2005.
- [14] S. Aly, N. Tsuruta, and R. Taniguchi, "Face recognition under varying illumination using Mahalanobis self-organizing map," *Artificial Life and Robotics*, Vol. 13, No. 1, pp. 298–301, 2008.
- [15] K. Kotani, Q. Chen, and T. Ohmi, "Face recognition using vector quantization histogram method," *IEEE 2002 International Conference on Image Processing, II-105–108*, 2002.
- [16] A. Gersho and R. M. Gray, "Vector quantization and signal compression," Kluwer Academic, 1992.
- [17] T. Nakayama, M. Konda, K. Takeuchi, K. Kotani, and T. Ohmi, "Still image compression with adaptive resolution vector quantization technique," *International Journal of Intelligent Automation and Soft Computing*, Vol. 10, No. 2, pp. 155–166, 2004.
- [18] P. J. Phillips, H. Wechsler, J. Huang, and P. Rauss, "The FERET database and evaluation procedure for face recognition algorithms," *Image and Vision Computing Journal*, Vol. 16, No. 5, pp. 295–306, 1998.
- [19] AT&T Laboratories Cambridge, "The database of faces," at: http://www.cl.cam.ac.uk/research/dtg/attarchive/face_database.html
- [20] T. Kohonen, "Self-Organizing Maps," Springer, USA, 1995.
- [21] Q. Chen, K. Kotani, F. Lee, and T. Ohmi, "Face recognition using codebook designed by code classification," 2006 IEEE International Conference on Signal and Image Processing, Hubli, India, pp. 397–401, December 2006.
- [22] Q. Chen, K. Kotani, F. Lee, and T. Ohmi, "A VQ-based fast face recognition algorithm using optimized codebook," *Proceedings of the 2008 International Conference on Wavelet Analysis and Pattern Recognition*, Hong Kong, pp. 298–303, August 2008.
- [23] Q. Chen, K. Kotani, F. Lee, and T. Ohmi, "VQ-based face recognition algorithm using code pattern classification and Self-Organizing Maps," *Proceedings of 9th International Conference on Signal Processing (ICSP'08)*, Beijing, pp. 2059–2064, October 2008.