Feature Vector Compression based on Least Error Quantization

Tomokazu Kawahara and Osamu Yamaguchi
Corporate Research and Development Center TOSHIBA Corporation
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki-shi, 212-8582, Japan
{tomokazu.kawahara, osamu1.yamaguchi}@toshiba.co.jp

Abstract

We propose a distinctive feature vector compression method based on least error quantization. This method can be applied to several biometrics methods using feature vectors, and allows us to significantly reduce the memory size of feature vectors without degrading the recognition performance. In this paper, we prove that minimizing quantization error between the compressed and original vectors is most effective to control the performance in face recognition. A conventional method uses non-uniform quantizer which minimizes the quantization error in terms of $L_2$-distance. However, face recognition methods often use metrics other than $L_2$-distance. Our method can calculate the quantized vectors in arbitrary metrics such as $L_p$-distance ($0 < p \leq \infty$) and the quantized subspace basis. Furthermore, we also propose a fast algorithm calculating $L_p$-distances between two quantized vectors without decoding them. We evaluate the performance of our method on FERET, LFW and large face datasets with LBP ($L_p$-distance), Mutual Subspace Method and deep feature. The results show that the recognition rate using the quantized feature vectors is as accurate as that of the method using the original vectors even though the memory size of the vectors is reduced to $1/5$ - $1/10$. In particular, applying our method to the state-of-the-art feature, we are able to obtain the high performance feature whose size is very small.

1. Introduction

Reduction of the memory size of features is one of the fundamental issues in biometric identification. Recently, there is a compelling need for multi-biometrics or multi-template authentication, because the concept of fusion in biometrics helps to keep high security, and the identification with several templates improves usability. While templates are increasing, it is desirable to minimize template size since the memory size of IC-Card is limited. In the case of face recognition, multiple templates are utilized for verification in video surveillance system which takes multiple faces from multiple cameras.

There is a face recognition method which uses a subspace as individual features. For example, in Mutual Subspace Method [30] a subspace which is generated from multiple feature vectors represents individual features. This representation is suboptimal from a statistical point of view because the subspace is generated by PCA [26]. Hence, additional memory size reduction is a hard problem. In addition, there is a recognition method whose similarity is calculated by $L_1$-distance instead of $L_2$-distance. Local Binary Pattern [2] is a famous feature in face recognition community and, its feature vector consists of histograms. LBP with $L_1$-distance is often more effective than $L_2$-distance. Therefore, a compression method which is independent of feature and distance is effective for face recognition.

In computer vision field, several small size features are proposed; Compressed Histogram of Gradients [4], elliptical regions with the gravity vector [20], learned binary codes [7, 15, 25, 28], and Regions with feature points [29]. Furthermore, compressed descriptors [10, 14] and codebook compression on the Bag of Features [6, 10, 17, 27] are proposed. Memory size of features of these methods are extremely small, but these features depend strongly on the recognition methods.

As a method-independent compression, we usually use statistical dimensionality reduction such as PCA [26], LDA [3], LPP [9], LCD [5]. These methods extract meaningful features from original high-dimensional vectors, based on the spatial bias of the feature vectors and their relationship. They are effective for compression of feature vectors, and can be applied to several methods which use feature vectors, but these method need training data. Furthermore, their compression is more efficient by combination of a essentially different method.

In the view of signal processing, Quantization, Discrete Cosine Transform (DCT) [21], and Vector Quantization (VQ) [8] are commonly used for image compression. DCT decomposes a signal into spatial frequency components derived from weighted sums of cosine harmonics,
and VQ constructs a codebook of representative vectors. These methods have a good property of image compression. However, DCT and dimensionally reduction methods have in common because DCT use a linear projection, and VQ needs a set of vectors. On the other hand, Quantization reduces the size of a vector by compressing a range of the values. This method is based only on the bias of values with ignoring the spatial structure. Therefore, Quantization does not use a linear projection and does not need training vectors. We compress feature vectors efficiently using non-uniform quantizer, when their values are distributed non-uniformly. In each vector, bias of its values are usually different. Therefore, instead of a common quantizer, we efficiently compress each vector with the non-uniform quantizer optimized for the vector, although we have to add another table to each quantized vector.

Among many quantization methods were proposed, Otsu proposed a quantization method [22], which uses non-uniform quantizer minimizing the quantization error in the least square sense for each gray scale image. This method does not need training data, and can be applied to several recognition methods using feature vectors. While \(L_2\)-distance is a reasonable choice for image compression, it is not always the case in face recognition, where metrics other than \(L_2\)-distance are also commonly used [1]. Furthermore, the relation between recognition performance and quantization error is not clear.

We propose a feature vector compression method from a viewpoint of recognition. We prove that minimizing quantization error is most effective to control the recognition performance, and generalize Otsu’s quantization method to handle arbitrary distance metrics, such as \(L_p\)-distance \((0 < p \leq \infty)\). Since our method enables to choose the appropriate distance metric with respect to a problem we address, better recognition performance is expected in comparison with that of Otsu’s quantization method. Figure 1 illustrates PCA, uniform quantizer and the proposed method applied to grid points on three-dimensional space. Furthermore, we also propose a fast algorithm for calculating distances without decoding the quantized vectors because distance calculation is not necessary in image compression, but necessary in pattern recognition. Finally, we show the effectiveness of the proposed method through experiments on FERET database [24], LFW [12] and a large scale face database. Our method gives equivalent recognition performance in comparison with that of the method using the original vectors even though the memory size is reduced to 1/5-1/10.

2. Relation between Quantization Error and Recognition Accuracy

We prove that minimizing quantization error of each feature vector is most effective to control the recognition accuracy of the method using quantized vectors. In this section, we analyze within-class and between-class scatter of compressed feature vectors, and apply this analysis to quantization.

2.1. Within-Class and Between-Class Scatter of compressed vectors

We describe, in section 2.1, the variance of compression error is the main factor to degrade recognition accuracy caused by generic feature vector compression.

We assume, in this paper, that original and compressed vectors are in the same space, and that a compression error \(\epsilon\) which is the difference between an original vector \(v\) and its compressed vector \(v'\), is normally distributed with a mean vector \(m\) and a covariance matrix \(\sigma^2I\), where \(I\) is the identity matrix:

\[
\epsilon = v - v' \sim N(m, \sigma^2 I), \tag{1}
\]

We compare the difference between original vectors and that between their compressed vectors;

\[
(v' - w') - (v - w) = (v' - v) - (w' - w) = \epsilon_v - \epsilon_w \sim N(0, 2\sigma^2 I), \tag{2}
\]

where \(v\) and \(w\) are original vectors, \(v'\) and \(w'\) are their compressed vectors, and \(\epsilon_v\) and \(\epsilon_w\) are their compression errors. From equation (2), the differences between these differences are normally distributed, and their mean and covariance matrix are 0 and \(2\sigma^2 I\), respectively. Therefore, their distances is normally distributed, and their mean and variance are 0 and \(2d\sigma^2\), where \(d\) is the dimension of vectors;

\[
||\epsilon_v - \epsilon_w|| \sim N(0, 2d\sigma^2), \tag{3}
\]
2.2. Quantization Error

We derive the distribution of distances between compressed vectors, its mean and variance. From equation (2), the distribution of distances between two compressed vectors \( \rho'(s) \) is the convolution of these two functions;

\[
\rho'(s) = (\rho \ast \epsilon)(s) = \int_{-\infty}^{\infty} \rho(t) \epsilon(t-s) dt, \tag{4}
\]

where \( \rho(s) \) is the distribution of distances between original vectors and \( \epsilon(s) \) is the distribution of the difference between compression errors defined by equation (3). From equations (4) and (3), the mean and the variance of \( \rho'(t) \) are the following;

\[
m(\rho') = m(\rho) + m(\epsilon) = m(\rho), \tag{5}
\]

\[
\sigma^2(\rho') = \sigma^2(\rho) + \sigma^2(\epsilon) = \sigma^2(\rho) + g(\sigma^2), \tag{6}
\]

where \( m(f) \) and \( \sigma^2(f) \) are the mean and the variance of function \( f \), respectively, and \( g(\sigma^2) \) is a monotone increasing function with respect to \( \sigma^2 \). From these equations, the distribution of distances between compressed vectors is depend only on the variance of compression errors and, their variance is a monotone increasing function with respect to the variance of compression errors.

We apply the the equations (5) and (6) to within-class and between-class scatter of original and compressed feature vectors (Figure 2). This shows that means of within-class and between-class distribution are same and their variances are increased by the variance of compression error. As a results, the variance of compression error is the main factor to degrade recognition accuracy caused by feature vector compression.

The minimum value of quantization error is the main factor to degrade recognition accuracy caused by feature vector compression.

In this paper, a \( d \)-dimensional quantized vector \( X' = (x'_1, \ldots, x'_d) \) of a original vector \( X = (x_1, \ldots, x_d) \) with \( N \)-level non-uniform quantizer \( Q_{T,A} \) whose thresholds are \( T = \{ t_1, \ldots, t_{N-1} \} \) and values \( A = \{ a_1, \ldots, a_N \} \) (Figure 3) is defined as follows;

\[
(x'_1, \ldots, x'_d) = (Q_{T,A}(x_1), \ldots, Q_{T,A}(x_d)), \tag{7}
\]

![Figure 3. N-level non-uniform quantizer whose thresholds are \( T = \{ t_1, \ldots, t_{N-1} \} \) and values \( A = \{ a_1, \ldots, a_N \} \)](image)

Quantization error is divided into two non-negative parts; the minimum part and the rest part. First, quantization error \( \epsilon(X, Q_{T,A}) \) is defined as follows;

\[
\epsilon(X, Q_{T,A}) = ||Q_{T,A}(X) - X||. \tag{9}
\]

This is a non negative function with respect to thresholds \( T \) and values \( A \). Therefore, it has the minimum value \( \epsilon_{\min}(X) \) and quantization error is divided into the minimum value and its rest \( \epsilon_{\text{rest}}(X, Q_{T,A}) \);

\[
\epsilon_{\min}(X) = \min_{T,A} \epsilon(X, Q_{T,A}). \tag{10}
\]

\[
\epsilon(X, Q_{T,A}) = \epsilon_{\min}(X) + \epsilon_{\text{rest}}(X, Q_{T,A}). \tag{11}
\]

The minimum \( \epsilon_{\min}(X) \) is depend only on a vector \( X \) and independent of thresholds \( T \) and values \( A \). On the other hand, the rest \( \epsilon_{\text{rest}}(X, Q_{T,A}) \) is a non-negative function with respect to thresholds \( T \), values \( A \) and a vector \( X \).

We consider those three distributions; Quantization error \( \rho(\epsilon_{\text{quant}}) \), its minimum \( \rho_{\min}(\epsilon) \) and rest \( \rho_{\text{rest}}(\epsilon) \). From the equation (11), the distribution of quantization error is the convolution of the others;

\[
\epsilon_{\text{quant}}(s) = \int_{-\infty}^{\infty} \epsilon_{\min}(s) \epsilon_{\text{rest}}(t-s) dt. \tag{12}
\]

From this equation, the variances of these distributions have the following relation;

\[
\sigma^2(\epsilon_{\text{quant}}) = \sigma^2(\epsilon_{\min}) + \sigma^2(\epsilon_{\text{rest}}). \tag{13}
\]
Therefore, the variance of quantization errors is the sum of that of minimum quantization errors and that of the rests. As a results, minimizing quantization error of each vector obtain the least variance of quantization errors.

3. Quantization with $L_2$-distance

We explain the quantization with $L_2$-distance based on Otsu’s quantization method [22]. This quantization is non-uniform quantizer which minimizes the quantization error in the least square sense for each vector. This minimization problem is solved by dynamic programing algorithm.

3.1. Problem Formulation

We prove that the minimum quantization of a vector is equivalent to the minimum clustering of the set which consists of vector values. We solve the equation (14).

$$\arg\min_{T,A} ||Q_{T,A}(X) - X||_2^2.$$  \hspace{1cm} (14)

where $|| \cdot ||_2$ is $L_2$-norm. First, we can assume $x_1 \leq \cdots \leq x_D$ without loss of generality because the quantization errors of $X$ and its permutation vector $X_{\text{perm}}$ are same values from the equation (15):

$$||Q_{T,A}(X) - X||_2^2 = \sum_{i=1}^d |x_i - Q_{T,A}(x_i)|^2$$  
$$= \sum_{j=1}^d |x_j - Q_{T,A}(x_j)|^2$$  
$$= ||Q_{T,A}(X_{\text{perm}}) - X_{\text{perm}}||_2^2$$  \hspace{1cm} (15)

Let $C_i = \{x_j | t_{i-1} \leq x_j < t_i \}$ be $i$-th cluster divided by thresholds $t_{i-1}$ and $t_i$, where $t_0$ is below $x_0$ and $t_d$ is above $x_d$. The quantizer $Q_{T,A}$ translates each element of $i$-th cluster $C_i$ to $a_i$ for $i = 1, \ldots, N$. The quantization error (14) is described by $C_i$ and $a_i$:

$$||Q_{T,A}(X) - X||_2^2 = \sum_{i=1}^N \sum_{x_i \in C_i} (a_i - x)^2,$$  \hspace{1cm} (16)

$$= \sum_{i=1}^N N_i ((a_i - \mu)^2 + \sigma_i^2).$$  \hspace{1cm} (17)

where $N_i$, $m_i$, and $\sigma_i^2$ are the number, mean, and variance of $C_i$, respectively. The right hand side of the equation (16) is equal to the clustering error, where the set $\bar{X} = \{x_1, \ldots, x_d\}$ is divided by $C_i$ whose average is $a_i$ ($i = 1, \ldots, N$) (Figure 4). The equation (17) shows that the clustering error of $C_i$ is minimum when each average $a_i$ is the mean of $C_i$. From these results, we solve the minimum clustering problem as follows:

$$\arg\min_{\bar{X} = C_1 \cup \ldots \cup C_N} \sum_{i=1}^N \sum_{x_i \in C_i} (m_i - x)^2.$$  \hspace{1cm} (18)

3.2. Search algorithm

We find the global minimum of clustering problem (18) with dynamic programing algorithm among all combinations of thresholds, whose number is $(d-1)C_{(N-1)}$. Otsu’s algorithm is based on the following overlapping problem: If the thresholds $t_1 \leq \cdots \leq t_{N-1}$ minimizes the clustering error of $\{x_1, \ldots, x_d\}$, then their subset $t_1 \leq \cdots \leq t_{N-2}$ minimizes the clustering error of $\{x_1, \ldots, x_{t_{N-1}}\}$ (Figure 5). We define $E^2(n, \delta)$ as the minimum clustering error of the set $\{x_1, \ldots, x_d\}$, where its clustering number is $n$. The solution of the original clustering problem is represented as $E^2(N, d)$. $E^2(n, \delta)$ satisfy the following equations for $\delta = 1, \ldots, d$ and $n = 1, \ldots, N$:

$$E^2(n, \delta) = \min_{n-1 \leq \delta \leq \delta - 1} \{E^2(n-1, t) + e^2(t+1, \delta)\}.$$  \hspace{1cm} (19)

$$E^2(1, \delta) = e^2(1, \delta).$$  \hspace{1cm} (20)

$$e^2(\delta, \delta') = \min_{\mu} \sum_{i=\delta}^{\delta'} (\mu - x_i)^2.$$  \hspace{1cm} (21)

The equation (21) is minimum when $\mu$ is equal to the mean of the cluster $\{x_\delta, \ldots, x_{\delta'}\}$.

Algorithm 1 denotes the global minimum search algorithm based on equation (19) and (20). $t(n, \delta)$ is the $n$-th threshold of $\{x_1, \ldots, x_d\}$. Each threshold $t_i$ of the solution are calculated by the equation (22) for $i = 1, \ldots, N-1$.

$$t_i = (x_{t(i-1, d)} + x_{t(i, d-1)})/2.$$  \hspace{1cm} (22)

This is a $O(Nd^2)$-time algorithm.
Algorithm 1 Search algorithm of minimum clustering

1: for all \( t = 1 \) to \( d \) do
2: \( E^2(1, t) \leftarrow e^2(1, t) \).
3: end for
4: for all \( n = 2 \) to \( N \) do
5: \( \text{for all } \delta = n - 1 \) to \( d \) do
6: \( \text{for all } t = n - 1 \) to \( \delta \) do
7: if \( E^2(n - 1, t) + e^2(t + 1, \delta) < E^2(n, \delta) \) then
8: \( E^2(n, \delta) \leftarrow E^2(n - 1, t) + e^2(t + 1, \delta) \)
9: \( t(n, \delta) \leftarrow t \)
10: end if
11: end for
12: end for
13: end for

4. Proposed Method

In this section, we describe three contributions. First, we extend Otsu’s quantization method to deal with any metrics. Second, we describe data structure of a quantized vector. Last, we propose a method for calculating the distance without decoding the quantized feature vectors.

4.1. Minimum \( L_p \)-Quantization Error

We extend the problem (18) to deal with \( L_p \)-distance:

\[
\arg\min_{T,A} \|Q_{T,A}(X) - X\|^p. \tag{23}
\]

We find the global minimum of clustering problem (23), using the algorithm in section 3.2 with the following \( e^p(\delta, \delta') \) instead of \( e^2(\delta, \delta') \):

\[
e^p(\delta, \delta') = \begin{cases} 
\min_{\mu} \sum_{i=1}^{d'} |\mu - x_i|^p & (0 < p < \infty), \\
\min_{\mu} \max_{d \leq d' \leq \delta} \{ |\mu - x_i| \} & (p = \infty),
\end{cases}
\]

(24)

For example, \( e^1(\delta, \delta') \) and \( e^{\infty}(\delta, \delta') \) are minimum when \( \mu \) is the median of a cluster \( \{x_\delta, \ldots, x_{d'}\} \) and \( (x_\delta + x_{d'})/2 \), respectively. Moreover, we can apply an arbitrary metric to the quantization using this extension.

4.2. Data Structure of A Quantized Vector

We propose data structure of a non-uniform quantized vector which minimizes the quantization error of each vector. Because this quantizer is optimized for each vector, each quantized vector needs to have its values. A \( N \)-level non-uniform quantized vector consists of not only a discrete sequence but also a value table:

\[
(x_1, \ldots, x_d) \rightarrow \begin{cases} 
(\{a_1, \ldots, a_N\} \text{ value table,}) & (25) \\
(\{n_1, \ldots, n_d\} \text{ discrete sequence.}) 
\end{cases}
\]

Since this discrete sequence consists of a bit sequence, memory size of a quantized vector is as follows;

\[
f \times N + \lceil \log_2(N) \rceil \times D \text{ (bit),} \tag{26}
\]

where \( f \) is the memory size of a continuous value, and \( \lceil n \rceil \) is the minimum integer that is not less than \( n \). Figure 6 is the graph of memory sizes of quantized and original vectors. This figure shows that the memory size of a quantized vector is much smaller than that of an original vector although it has its value table.

4.3. Distance Calculation

We propose calculation of the distance between quantized vectors without decoding.

The \( L_p \)-distance between quantized vectors \( v' \) and \( w' \) is calculated:

\[
||v' - w'||_p = \sum_{i=1}^{d} |a_{n_i} - \tilde{a}_{\tilde{n}_i}|^p, \tag{27}
\]

where \( \{a_i, (n_j)\} \) and \( \{\tilde{a}_i, \tilde{n}_j\} \) are the data structure of \( v' \) and \( w' \), respectively \( (i = 1 \ldots N, j = 1 \ldots d) \). This calculation spend more time than the calculation of the distance between original vectors because conversion from a bit sequence to discrete values is added.

We propose calculation which consists of following two steps: We generate the table \( \{A_{i,j}\}_{i,j=1\ldots n} \) which is all combinations of values on value tables, and add the corresponding values on the table (Equation (28), (29)).

\[
A_{i,j} = |a_i - \tilde{a}_j|^p \quad (i, j = 1 \ldots n), \tag{28}
\]

\[
||v' - w'||_p = \sum_{i=1}^{d} A_{n_i, \tilde{n}_i}, \tag{29}
\]

In the second step, our calculation consists of only conversion to discrete values and addition of values of the table. This calculation is as fast as the calculation of \( L_p \)-distance between original vectors.

To evaluate the process time of the proposed calculation, we measured \( L_2 \)-distance calculation time of 1000 pairs of quantized vectors using this method and that of 1000 pairs
of original vectors in the environment of Intel Core 2 Extreme PC with 3GHz CPU. Figure 7 shows the total time of calculation of each 1000 pairs. From this figure, the proposed calculation was as fast as calculation of the distance between original vectors regardless of their dimension.

5. Experiments

We applied the proposed method to Local Binary Pattern [2] with several metrics, Nearest Neighbor, Subspace Method on a large face database and a deep feature of VGG-Face CNN descriptor [23].

5.1. Evaluation of Distances

We evaluate the proposed method applied to a method using $L_p$-distance ($p \neq 2$).

We used fa, fb, fc, dup I and dup II in FERET [24] database. "fa" set, used as a gallery set, contains frontal images of 1,196 people. "fb" set contains 1,195 images and their subjects were asked for an alternative facial expression than in the fa photograph. "fc" set contains 194 images and their photos were taken under different lighting conditions. "dup I" set contains 722 images and their photos were taken later in time. "dup II" set contains 234 images is a subset of the dup I set containing those images that were taken at least a year after the corresponding gallery image. The evaluation index is Correct Match Rate.

We describe the generation of LBP feature vector in this evaluation. The faces and the feature points were manually located. To extract robust features for head pose, we applied 3D normalization [16]. We cropped a face whose size is $96 \times 96$ pixels from the normalized image, and translate the cropped face to Local Binary Pattern (LBP) [2]. We divided the LBP pattern face to $8 \times 8$ rectangular regions, and concatenated histograms which count each pattern on each region, to generate a feature vector. The dimension of the feature vector is $16384 = 256 \times 8 \times 8$.

First, we show $L_1$-distance is better for the LBP feature than $L_2$-distances. The results for the LBP feature vector with $L_1$, $L_2$, and $L_\infty$-distance are shown in Table 1. The LBP feature vector with $L_1$-distance is better than that with $L_2$-distance with respect to “Mean”. In particular, $L_1$-distance is more better than $L_2$-distance in the case “fb” and “dup II”. On the other hand, $L_1$-distance is worse than $L_2$-distance in the case “fc” and “dup II”, however the differences are small. Therefore, $L_1$-distance is better for the LBP feature than $L_2$ and $L_\infty$-distances.

We applied the proposed method to the LBP feature vector with $L_1$-distance. The results for the LBP feature vector $L_p$-compressed ($p = 1, 2, \infty$) with level 16 shown in Table 2. The $L_1$-compression is better than that with $L_2$-compression with respect to “dup I”, “dup II” and “Mean”. In the case “fb” and “fc”, $L_2$-compression is better than $L_1$-compression, however the differences are small. Therefore, $L_1$-compression is better for the LBP feature with $L_1$-distance than $L_2$ and $L_\infty$-compression.

The size of an original LBP feature vector is 64 KByte and that of the compression feature vector (level 16) is 8 KByte. Our method reduces size of a feature vector to 1/8.

5.2. Evaluation of Compressing Vectors with PCA and Compressing Subspaces

We evaluated the proposed method applied to a feature vector with other dimensional reduction scheme, and a complex feature, such as a basis of subspace.

We prepared the face database consisted of 21771 individuals, which consisted of public face databases and originally collected images. We used 21771 gallery images and 908 probe images in the face recognition experiments. The faces and the feature points were manually located. To extract robust features for head pose and illumination variation, we applied 3D normalization [16] and preprocessing [19] to these face images. The dimension of feature vectors was 256.
We evaluated the proposed method with $L_p$-distance ($p = 1, 2$) with respect to memory size of a feature vector and recognition accuracy. We applied the 8-level proposed method with $L_p$-distance ($p = 1, 2$) and 8-level uniform quantization to Nearest Neighbor method (NN) and NN with PCA using these distances.

The projections of PCA were generated from the gallery images and their dimensions were 80 on $L_1$-NN and 130 on $L_2$-NN, which were the best dimension for this database. Figure 8 shows the results of $L_p$-NN and those of $L_p$-NN with PCA ($p = 1, 2$), respectively. The left and right horizontal axes of these figures indicate memory size of a feature vector and Equal Error Rate (EER), respectively. This rate is the probability that false acceptance rate (FAR) equals the false rejection rate (FRR).

From Figure 8, the proposed method reduced memory size of a feature vector without sacrificing the recognition accuracy. The memory size of a feature vector of uniform quantization was smaller than those of proposed methods, but its EER was much larger than those of proposed methods. In particular, when the distance of proposed method and that of NN were same, its EER was the least. As a result, our method was more efficient than Otsu’s quantization, which uses only $L_2$-distance. In addition, using PCA and our method, we were able to reduce more memory size of a feature vector than using only PCA.

We applied the proposed method to another face recognition method the Whitened Mutual Subspace Method (WMSM) [13]. This method uses a subspace as an individual feature, and calculate a similarity between two subspaces using inner products of their orthonormal basis. A inner product of normal vectors is related to $L_2$-distance:

$$||x-y||_2^2 = ||x||_2^2 + ||y||_2^2 - (x, y) = 2 - (x, y).$$

(30)

where, $x, y$ are normal vectors and $(x, y)$ is their inner product. Therefore, we applied the proposed method with $L_2$-distance to orthonormal bases of WMSM. The parameters of WMSM were same in [13]. Figure 9 shows the results of experiments on the large face database with original WMSM and WMSM to which we applied the proposed methods whose levels were 2, 4, 8, and 16.

The results of this experiment was that EER of original WMSM was 4.4 % and that of WMSM with the 16-level proposed method were 4.5 %. Hence, the proposed method gave equivalent recognition accuracy in comparison with that of the method using the original bases even though the memory size is reduced to 1/5. As a result, the proposed method reduced memory size of complex features, such as a basis of subspace without degrading the recognition performance.

5.3. Evaluation of Compressing Deep Features

We applied this method to a deep feature of VGG-Face CNN descriptor [23]. This is one of state-of-the-arts face
We evaluated “fc6” feature which was learnt by CNN and the feature with \( L_2 \)-compression (level = 2, 4, 8, 16) on FERET. Table 3 shows these results. SLBFLE [18] is one of the state-of-the-art feature descriptors. fc6 feature and fc6 with our compression (level = 2, 4, 8, 16) are better than SLBFLE. The dimension of fc6 feature is 4096 and its size is 16.4 KByte. The size of the compression feature with level 2 is 520 Byte. Applying our method to a deep feature, we obtain a high performance face feature whose size is very small.

We also evaluated “fc6” feature with \( L_2 \)-compression on the “deep funneled” images [11] in the LFW dataset [12]. Table 4 shows these results. The average verification rate of the original fc6 feature is equal to that of fc6 with level 16 compression. Applying our compression with level 16, we obtain the state-of-the-art feature whose size is 2.1 KByte.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Level</th>
<th>Accuracy (± standard error)</th>
<th>Size (KByte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG-Face fc6</td>
<td>16</td>
<td>96.62 ± 3.1</td>
<td>16.4</td>
</tr>
<tr>
<td>Compressed fc6</td>
<td>2</td>
<td>93.08 ± 3.7</td>
<td>0.5</td>
</tr>
<tr>
<td>Compressed fc6</td>
<td>4</td>
<td>96.00 ± 3.5</td>
<td>1.0</td>
</tr>
<tr>
<td>Compressed fc6</td>
<td>8</td>
<td>96.52 ± 3.1</td>
<td>1.6</td>
</tr>
<tr>
<td>Compressed fc6</td>
<td>16</td>
<td>96.62 ± 3.1</td>
<td>2.1</td>
</tr>
</tbody>
</table>

Figure 9. Evaluation of WMSM with proposed methods whose level were 2, 4, 8, 16, on the large scale face database. The right horizontal axis A number in this graph denotes memory size of a basis of each method and its EER. EER of original WMSM was 4.4%. That of WMSM with 16-level quantization were 4.5%, and its memory size was reduced to 1/5.

Table 3. Correct Match Rate(%) of VGG-Face CNN descriptor (using \( L_2 \)-distance) with \( L_2 \)-compression (level = 2, 4, 8, 16).

<table>
<thead>
<tr>
<th>Feature</th>
<th>fb</th>
<th>fc</th>
<th>dup I</th>
<th>dup II</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLBFLE (R=4)</td>
<td>99.9</td>
<td>100.0</td>
<td>95.2</td>
<td>92.7</td>
</tr>
<tr>
<td>VGG-Face fc6</td>
<td>99.9</td>
<td>100.0</td>
<td>98.2</td>
<td>97.0</td>
</tr>
<tr>
<td>Compressed fc6</td>
<td>99.9</td>
<td>100.0</td>
<td>97.4</td>
<td>94.9</td>
</tr>
<tr>
<td>Compressed fc6</td>
<td>99.9</td>
<td>100.0</td>
<td>97.9</td>
<td>96.6</td>
</tr>
<tr>
<td>Compressed fc6</td>
<td>99.9</td>
<td>100.0</td>
<td>98.2</td>
<td>97.0</td>
</tr>
</tbody>
</table>

6. Conclusion

We have analyzed the relation between recognition performance and quantization error, and have proposed a distinctive feature vector compression method using least error quantization calculated with arbitrary distance metrics such as \( L_p \)-distance \((0 < p \leq \infty)\). Furthermore, we have proposed effective algorithms for distance calculation of quantized vectors without decoding. To evaluate this method, we applied this method to LBP, NN, NN with PCA, WMSM and deep feature learnt by CNN and experimented on FERET, LFW and a large face database. These results show that the proposed method can be applied to several recognition methods and significantly reduces the memory size of feature vectors without degrading the recognition performance. In particular, applying our method to the state-of-the-art feature, we are able to obtain the high performance face feature whose size is very small.

References


