Effect of Fuzzy and Crisp Clustering Algorithms to Design Code Book for Vector Quantization in Applications

Yukinori Suzuki, Hiromu Sakakita and Junji Maeda
Department of Information and Electronic Engineering, Muroran Institute of Technology,
27-1 Mizumoto-cho, Muroran, Hokkaido 050-8585, Japan

Keywords: Fuzzy and Crisp Clustering Algorithms, Vector Quantization, Code Book Design.

Abstract: Image coding technologies are widely studies not only to economize storage device but also to use communication channel effectively. In various image coding technologies, we have been studied vector quantization. Vector quantization technology does not cause deterioration image quality in a high compression region and also has a negligible computational cost for image decoding. It is therefore useful technology for communication terminals with small payloads and small computational costs. Furthermore, it is also useful for biomedical signal processing: medical imaging and medical ultrasound image compression. Encoded and/or decoded image quality depends on a code book that is constructed in advance. In vector quantization, a code book determines the performance. Various clustering algorithms were proposed to design a code book. In this paper, we examined effect of typical clustering (crisp clustering and fuzzy clustering) algorithms in terms of applications of vector quantization. Two sets of experiments were carried out for examination. In the first set of experiments, the learning image to construct a code book was the same as the test image. In practical vector quantization, learning images are different from test images. Therefore, learning images that were different from test images were used in the second set of experiments. The first set of experiments showed that selection of a clustering algorithm is important for vector quantization. However, the second set of experiments showed that there is no notable difference in performance of the clustering algorithms. For practical applications of vector quantization, the choice of clustering algorithms to design a code book is not important.

1 INTRODUCTION

Transmission bandwidths of the Internet have increased remarkably, and images, videos, voices, musics and texts can now be transmitted instantly to every corner of the world. Immeasurable quantities of data are being transmitted around the world. Although new technologies have provided larger transmission bandwidths, the demand for transmission capacity continues to outstrip the capacity implemented with current technologies. The demand for communication capacity shows no sign of slowing down. Data compression technology is therefore important for effective use of communication channels. Since image and video data transmitted through the Internet occupy a large communication bandwidth, various coding methods have been proposed. There are two categories of coding methods: lossless coding and lossy coding. In lossless coding, the original image and video can be recovered from the compressed image and video completely. Huffman coding is the most widely used type of lossless coding. In lossy coding, on the other hand, the original image and video cannot be recovered from the compressed image and video. In lossy coding, JPEG (Joint Photographic Experts Group) is used as a defect standard for still image coding and MPEG (Moving Picture Experts Group) is used as a defect standard for video coding. The compression ratio of lossless coding is much smaller than that of lossy compression (Sayood, 2000; Gonzalez, 2008).

We have been studying vector quantization for image coding (Miyamoto, 2005; Sasazaki, 2008; Matsumoto, 2010; Sakakita, 2014). For vector quantization, we have to design a code book in advance. It needs much computational cost to design a code book. However, once we obtained the code book, encoding an image is to look for the nearest code vector in the code book. For decoding, it is only simple table look-up of the code vectors from the code book and therefore computational cost is negligible. Fujibayashi and Amerijckx showed that PSNR (peak-signal-to-noise ratio) gradually decreases as compression ratio increases in the case of vector quantiza-

Suzuki Y., Sakakita H. and Maeda J..
Effect of Fuzzy and Crisp Clustering Algorithms to Design Code Book for Vector Quantization in Applications.
DOI: 10.5220/0005207501980205
In *Proceedings of the International Conference on Bio-inspired Systems and Signal Processing* (BIOSIGNALS-2015), pages 198-205
ISBN: 978-989-758-069-7

tion while PSNR rapidly decreases as compression ratio increases in the case of JPEG (Fujibayashi, 2003; Amerijcks, 1998). In (Fujibayashi, 2003; Amerijcks, 1998), it was shown that an *PSNR* of the image compressed with JPEG rapidly decreases compared with that in the case of vector quantization when the compression ratio is larger than approximately 28. These results were supported by results reported by Laha et al. (Laha, 2004). Vector quantization is therefore considered to be appropriate for image coding to achieve a high compression ratio. A high compression ratio of an image decreases the payloads of communication terminals. Furthermore, decoding by table look-up of the code vectors decreases computational costs of the terminals. In this sense, vector quantization is useful technology for communication terminals with small payloads and small computational costs. We call these conditions SPSC (small payload and small compression cost). Furthermore, vector quantization is useful technology for medical image processing and medical ultrasound image compression (Hosseini, 2012; Hung, 2011; Nguyen, 2011). Small computational cost was allowed using vector quantization for medical image processing.

For vector quantization, we have to initially construct a code book. Fig. 1 shows a conceptual diagram of the vector quantization. At the source side, an image is divided into square blocks of $\tau \times \tau$ such as 4×4 or 8×8 pixels. Each block is treated as a vector in a 16- or 64-dimensional space. For encoding an image, each vector made by a block of pixels is compared with a code vector in a code book. The nearest code vector is chosen for the vector and the index of the code vector is assigned to the vector. Selection of a code vector is carried out for all vectors. This encoding process generates an index map as shown in Fig. 1. The index map consists of integer numbers. For the code book consisting of 256 code vectors, indexes take values from 1 to 256. These indexes are compressed using entropy coding and sent to a destination via a communication channel. At the destination, the compressed integer numbers are decoded to obtain the index map. We look for the code vector corresponding to the indexes in the index map using a code book and decode the image sent from the source side. As stated before, this is a table look-up process and consequently computational cost is negligible. Since the quality of a decoded image depends on the code vectors in a code book, code book design is important for vector quantization. A code book is constructed by the following procedure. First, the size of a code book, which is the number of code vectors is determined. The learning images are prepared to compute code vectors. Each of the images is divided

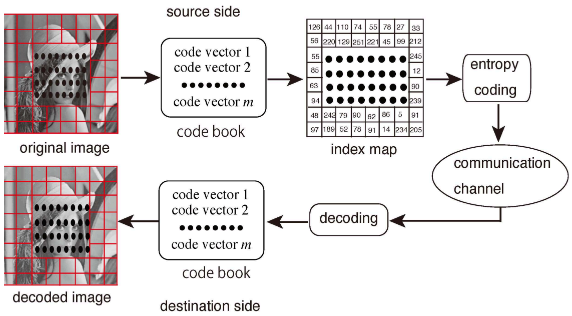

Figure 1: A conceptual diagram encoding/decoding with VQ.

into rectangular blocks such as 4×4 or 8×8. Each block is treated as a vector and it is called a learning vector. All learning vectors are classified into classes. Prototype vectors generated through classification are code vectors.

Clustering, self-organizing feature map (SOFM) and evolutionary computing were used to design a code book. In these methods, clustering algorithms were widely used, because they were easy to implement. In hard clustering, since each learning vector belongs to only one cluster, the membership value is unitary. On the other hand, each vector belongs to several clusters with different degrees of membership in fuzzy clustering (Patane, 2001). The *k* means algorithm, also called LBG (Linde-Buzo-Gray) algorithm (Linde, 1980) is the most popular algorithm. The main problem in clustering algorithms is that the performance is strongly dependent on the choice of initial conditions and configuration parameters. Patane and Russo (Patane, 2001) improved the LBG algorithm and named it the enhanced LBG algorithm. The fuzzy *k* means algorithm is the most popular fuzzy clustering algorithm (Bezdek, 1987). There are fuzzy clustering algorithms to implement transition from fuzzy to crisp. The typical algorithm is fuzzy learning vector quantization (FLVQ) (Tsao, 1994; Tsekouras, 2008). In the FLVQ algorithm, a smooth transition from fuzzy to crisp mode is accomplished to manipulate the fuzziness parameter.

A self-organizing feature map (SOFM) shows interesting properties to design a code book (Amerijcks, 1998; Laha, 2004; Tsao, 1994). Those are topology preservation and density mapping. Vectors near the input space are mapped to the same node or a nearby node in the output space by the property of topology preservation. After training the input vectors, the distribution of weight vectors of the nodes reflects the distribution of training vectors in the input space by the property of density mapping. By these two properties, there are more code vectors in a region with a high density of training vectors. For a code book design using evolutionary algorithms, fuzzy particle swarm optimization (FPSO), quantum particle swarm optimization (QPSO), and firefly (FF) algorithms have been recently been reported as successful algorithms (Feng, 2007; Wang, 2007; Horn, 2012). These algorithms are all swarm optimization algorithms. PSO is an evolutionary computation technique in which each particle represents a potential solution to a problem in multi-dimensional space. A fitness evaluation function was defined for designing a code book. The population of particles is moving in the search space and every particle changes its position according to the best global position and best personal position. These algorithms were developed to construct a near-optimal code book for vector quantization to avoid getting stuck at local minima in the finally selected code book. The FF algorithm was inspired by social behavior of fireflies and showed the best performance among swarm optimization algorithms (Horn, 2012). Another successful swarm-based algorithm is the honey bee mating optimization (HBMO) algorithm, which was inspired by the intelligent behavior of bees in a honey bee colony (Abbass, 2001). The algorithm is based on a model that simulates the evolution of honey bees starting with a solitary colony to the emergence of an eusocial colony.

To design a code book for vector quantization, clustering algorithms (crisp and fuzzy) are simple to reduce computational costs and they therefore were widely used. However, previous studies focused on reduction of the average distortion error of coding images (Patane, 2001; Tsekouras, 2008; Kayayiannis, 1995; Tsekouras, 2005; Tsolakis, 2012). Those were basically studies on clustering algorithms, not on image coding, and the results of those studies were not sufficient to consider practical applications of clustering algorithms for image coding. In this study, we examined the effect of clustering algorithms for designing a code book in terms of practical applications. For examination, we prepared two types of images: learning images and test images. A code book was constructed using the learning images and test images were used to examine the performance of the code book for vector quantization. In most previous studies, the learning images were the same as the test images (Amerijcks, 1998; Patane, 2001; Tsekouras, 2008; Feng, 2007; Horn, 2012; Kayayiannis, 1995; Tsekouras, 2005; Tsolakis, 2012). For example, a code book was constructed with the Lenna image as a learning image, and the same Lenna image was also used as the test image. However, as shown in Fig. 1, an image is encoded using a code book that has to be prepared in advance for practical usage of vector quantization. The learning images to construct a code book are different from images to be encoded. For that reason, we examined clustering algorithms under the condition of learning images being different from test images. There are many clustering algorithms (Jain, 1999). We chose two crisp clustering algorithms: k means algorithm and ELBG algorithm (Patane, 2001). They are simple algorithms and widely used to design a code book (Tsekouras, 2008; Tsekouras, 2005; Tsolakis, 2012). For fuzzy clustering algorithms, we chose fuzzy k means and fuzzy learning vector quantization (Bezdek, 1987; Tsao, 1994). They are also frequently employed to design code book for vector quantization (Tsekouras, 2008; Tsolakis, 2012). We carried out two sets of experiments. In the first sets of experiments, we examined four clustering algorithms using Lenna image. The Lenna image was used both learning and test images. In the second sets of experiments, we examined the clustering algorithms under the condition that learning images were different from test images. We have carried out these experiments in (Sakakita, 2014). However, test images used in the experiments were different from test images employed in (Sakakita, 2014) to confirm the performance of clustering algorithms.

The paper is organized as follows. Four representative clustering algorithms used in image coding are briefly described in section 2. Results of examination for vector quantization by the four clustering algorithms are presented in section 3, and conclusions are given in section 4.

2 CLUSTERING ALGORITHMS AND CODE BOOK DESIGN

A learning image $\mathcal{X}$ is divided into square blocks of $\tau \times \tau$ pixels, and each block is treated as a learning vector that is represented as $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, where $\mathbf{x}_i \in R^{\tau\times\tau}$. We design a code book from these learning vectors, which consists of code vectors such as $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_k\}$, where $\mathbf{c}_j \in R^{\tau\times\tau}$. To design a code book, we organize these learning vectors into clusters using a clustering algorithm. Code vectors are computed so as to minimize discrepancy between learning vectors and code vectors by clustering algorithms. The following average distortion measure is frequently employed for clustering algorithms (Patane, 2001).

$$D_{ave} = \frac{1}{n}\sum_{i=1}^{n} \min_{\mathbf{c}_j \in C} d(\mathbf{x}_i, \mathbf{c}_j), \tag{1}$$

where $d(\mathbf{x}_i, \mathbf{c}_j)$ is the distance between $\mathbf{x}_i$ and $\mathbf{c}_j$. We used the squared Euclidean norm for the distance.

D_{ave} is also called MQE (mean quantization error).

2.1 k-Means Clustering Algorithm

In the k-mean clustering (KMC) algorithm, each learning vector is assigned to a certain cluster by the nearest neighbor condition (Kayayiannis, 1995; Jain, 1999). The learning vector $\mathbf{x}_i$ is assigned to the jth cluster when $d(\mathbf{x}_i, \mathbf{c}_j) = \min_{\mathbf{c}_j \in \mathcal{C}} d(\mathbf{x}_i, \mathbf{c}_j)$ is true. The following membership function is derived from the nearest neighbor condition (Kayayiannis, 1995)

$$\mathbf{u}_j(\mathbf{x}_i) = \begin{cases} 1 \text{ if } d(\mathbf{x}_i, \mathbf{c}_j) = \min_{\mathbf{c}_j \in \mathcal{C}} d(\mathbf{x}_i, \mathbf{c}_j) \\ 0 \text{ otherwise.} \end{cases} \tag{2}$$

$\mathbf{c}_j$ is updated so as to minimize a distortion function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} u_j(\mathbf{x}_i) \left\| \mathbf{x}_i - \mathbf{c}_j \right\|^2, \tag{3}$$

where $\left\| \mathbf{x}_i - \mathbf{c}_j \right\|$ is Euclidean norm. The minimization of (3) is achieved by updating $\mathbf{c}_j$ according to the the equation

$$\mathbf{c}_j = \frac{\sum_{i=1}^{n} u_j(\mathbf{x}_i)\mathbf{x}_i}{\sum_{i=1}^{n} u_j(\mathbf{x}_i)}. \tag{4}$$

where $j = 1, 2, \cdots, k$. For experiments using KMC, we set a convergence condition to terminate repetition as

$$\frac{|D_{ave}(\nu - 1) - D_{ave}(\nu)|}{D_{ave}(\nu)} < \varepsilon, \tag{5}$$

where ν is the number of iterations and $\varepsilon = 10^{-4}$.

2.2 LBG and Enhanced LBG Algorithms

In the LBG (Linde-Buzo-Gray) algorithm, there are two methods of initialization: random initialization and initialization by splitting (Patane, 2001; Linde, 1980). In random initialization, the initial code vectors are randomly chosen from $\mathcal{X}$. This algorithm is KMC algorithm. On the other hand, initialization by splitting requires the number of code vectors to be a power of 2. The clustering procedure starts with only one cluster. The cluster is recursively split into two distinct clusters. After splitting the clusters up to predetermined number, the code vectors of respective clusters are computed. We obtain the code book of $\mathcal{C}^\nu$, where ν stands for the number of times the clusters were split. The learning vector $\mathbf{x}_i (i = 1, 2, \cdots, n)$ is assigned to the jth cluster according to the nearest neighbor condition, $d(\mathbf{x}_i, \mathbf{c}_j) = \min_{\mathbf{c}_j \in \mathcal{C}^\nu} d(\mathbf{x}_i, \mathbf{c}_j)$. The membership function is also defined as

$$\mathbf{u}_j(\mathbf{x}_i) = \begin{cases} 1 \ \text{ if } d(\mathbf{x}_i, \mathbf{c}_j) = \min_{\mathbf{c}_j \in \mathcal{C}^\nu} d(\mathbf{x}_i, \mathbf{c}_j) \\ 0 \ \text{ otherwise.} \end{cases} \tag{6}$$

The code vector $\mathbf{c}_j \in \mathcal{C}^\nu$ is updated by formula (4) using the above membership function.

The enhanced LBG (ELBG) algorithm was proposed by Patane and Russo (Patane, 2001). Patane and Russo revealed a drawback of the LBG algorithm. The LBG algorithm finds local optimal code vectors that are often far from an acceptable solution. In the LBG algorithm, a code vector moves through a contiguous region at each iteration. Thus, a bad initialization could lead to the impossibility of finding a good solution. To overcome this drawback of the LBG algorithm, they proposed a code vector shift from a small cluster to a larger one. They introduced the idea of utility index of a code vector. After the LBG algorithm, the ELBG algorithm identifies clusters with low utility and attempts to shift a code vector with low utility close to a code vector with high utility. The clusters with low utility are merged to the adjacent clusters. For our computational experiments, distortion to terminate the repetition was computed by (5) and ε was set to 10^{-4}.

2.3 Fuzzy k-Means Clustering Algorithm

The k-means clustering algorithm assigns each learning vector to a single cluster based on a hard decision. On the other hand, the fuzzy k-means clustering (FKM) algorithm assigns each learning vector to multiple clusters with membership values being between zero and one (Bezdek, 1987; Kayayiannis, 1995; Tsekouras, 2005). The learning vectors are assigned to clusters so as to minimize the objective function

$$J_m = \sum_{j=1}^{k} \sum_{i=1}^{n} u_j(\mathbf{x}_i)^m \left\| \mathbf{x}_i - \mathbf{c}_j \right\|^2 \tag{7}$$

under the following constraints:

$$0 < \sum_{i=1}^{n} u_j(\mathbf{x}_i) < n \tag{8}$$

$$\sum_{j=1}^{k} u_j(\mathbf{x}_i) = 1. \tag{9}$$

$1 < m < \infty$ controls the fuzziness of the clustering and it is given in advance. If m is equal to 1, it is crisp clustering. Minimization of (7) results in membership function update such that

$$u_j(\mathbf{x}_i) = \frac{1}{\sum_{l=1}^{k} \left(\frac{d(\mathbf{x}_i, \mathbf{c}_j)}{d(\mathbf{x}_i, \mathbf{c}_l)} \right)^{\frac{2}{m-1}}}, \tag{10}$$

where $d(\mathbf{x}_i, \mathbf{c}_j) = \left\| \mathbf{x}_i - \mathbf{c}_j \right\|^2$. The distance becomes zero, it is replaced by one to avoid zero division in our experiments. For update of the membership function by (10), the code vectors were renewed as

$$\mathbf{c}_j = \frac{\sum_{i=1}^{n} u_j(\mathbf{x}_i)^m \mathbf{x}_i}{\sum_{i=1}^{n} u_j(\mathbf{x}_i)^m} (j = 1, 2, \cdots, k). \quad (11)$$

For our computational experiments, m was set to 1.2. The repetition of the FKM algorithm terminates when ε of (5) is less than 10^{-4}.

2.4 Fuzzy Learning Vector Quantization

Transition from a fuzzy condition to a crisp condition is achieved by gradually decreasing the fuzziness parameter m in fuzzy learning vector quantization (FLVQ) (Tsao, 1994; Tsekouras, 2008). The objective function and the constraints are the same as those of the FKM algorithm. A fuzziness parameter is decreased from the initial value m_0 to m_f according to

$$m(t) = m_0 - [t(m_0 - m_f)]/t_{\max}, \quad (12)$$

for $t = 0, 1, 2, \cdots, t_{max}$. m_0 and m_f are the initial and final values, respectively and t_{max} is the maximum number of iterations. The membership function is updated for $t = 1, 2, \cdots, t_{max}$ as

$$u_j^t(\mathbf{x}_i) = \left[\sum_{l=1}^{k} \left(\frac{d(\mathbf{x}_i, \mathbf{c}_j)}{d(\mathbf{x}_i, \mathbf{c}_l)} \right)^{\frac{2}{(m(t)-1)}} \right]^{-m(t)}, \quad (13)$$

where $d(\mathbf{x}_i, \mathbf{c}_j) = \left\| \mathbf{x}_i - \mathbf{c}_j \right\|^2$. The distance becomes zero, it is replaced by one to avoid zero division in our experiments. The code vectors were also evaluated by the equation

$$\mathbf{c}_j^t = \frac{\sum_{i=1}^{n} u_j^t(\mathbf{x}_i) \mathbf{x}_i}{\sum_{i=1}^{n} u_j^t(\mathbf{x}_i)} \quad (14)$$

We set a convergence condition to terminate repetition as

$$\sum_{j=1}^{k} \left\| c_j^{t-1} - c_j^t \right\|^2 < \varepsilon \quad (15)$$

where $\varepsilon = 10^{-4}$. For our computational experiments, we used the following parameters: $m_0 = 1.5, m_f = 1.001$, and $t_{max} = 100$.

3 EFFECT OF CLUSTERING ALGORITHMS FOR VECTOR QUANTIZATION

To examine clustering algorithms, we carried out two sets of experiments.[1] In the first set of experiments, we examined four clustering algorithms using the Lenna image, which is 256×256 in size and an 8-bit grayscale image. This image was used as both the learning image and the test image. The Lenna image was segmented into 4×4 blocks in size. Each block was treated as a learning vector with 16 dimensions. There were 4096 learning vectors to construct a code book. We designed a code book consisting of 256 code vectors. It must be the minimum number of code vectors for us to keep acceptable image quality. In this sense, the number of 256 code vectors is suitable for comparative studies of clustering algorithms. 4×4 blocks in size means that each block containing $4 \times 4 = 16$ pixels is represented by 8 bits (256 code vectors). The compression rate is therefore $8/16 = 0.5$ *bits per pixel (bpp)*. The four clustering algorithms tested were the KMC algorithm, ELBG algorithm, FKM algorithm and FLVQ algorithm. The performance of each of the four clustering algorithms was examined in terms of *PSNR*. The *PSNR* is computed as

$$PSNR = 10 \log_{10} \left(\frac{PS^2}{MSE} \right) (\text{dB}), \quad (16)$$

where $PS = 255$. *MSE* is the mean square error between the original image and the decoded image. Since we carried out five trials for each clustering algorithm to design code books, five code books were constructed for one clustering algorithm. Thus, a total of 20 code books were constructed. *PSNR* was computed to examine the performance of each algorithm. Table 1 shows the results of these experiments. Among the four clustering algorithms, KMC showed the smallest average *PSNR* (29.42 dB) and ELBG showed the largest average *PSNR* (30.40 dB). The difference between those two values is 0.98 dB, which is a large difference for image quality. The results indicate that selection of a clustering algorithm is important for designing a code book. Tsekouras et al. (Tsekouras, 2008) reported results of an experiment in which they used the Lenna image with a size of 512×512 pixels and 8-bit gray scale. When the number of code vectors was 256, the difference in *PSNR* between KMC and their proposed clustering algorithm was 2.653 dB. Tsolakis et al. (Tsolakis,

[1] Images used in the experiments were images from CVG-UGR-Image database (http://decsai.ugr.es/cvg/dbimagenes/index.php)

Table 1: The *PSNR*s (dB) for four clustering algorithms (Lenna). *PSNR*s were rounded at the second decimal places.

trial	KMC PSNR	ELBG PSNR
1	29.47	30.37
2	29.47	30.37
3	29.52	30.42
4	29.35	30.39
5	29.30	30.42
average	29.42	30.40

trial	FKM PSNR	FLVQ PSNR
1	30.07	30.24
2	29.99	30.15
3	29.99	30.07
4	30.07	30.06
5	29.88	30.01
average	30.00	30.10

2012) carried out experiments using the same Lenna image as that above, and the difference in *PSNR* between the LBG algorithm and their proposed algorithm ($\theta = 0.5$) was 2.5429 dB. Their experiments also demonstrated that selection of a clustering algorithm was important for designing a code book of high quality.

A code book must be prepared in advance for practical application of image coding using vector quantization. This means that the learning images to construct a code book are different from images to be coded. It is therefore necessary to examine clustering algorithms under the condition of learning images to construct a code book being different from test images. In other words, the performance of vector quantization must be examined not only for specific learning images but also for previously unseen test images (Lazebnik, 2009). In the second set of experiments, we therefore prepared 20 learning images that were consisted of some images as shown in Fig. 2. Each image is 256×256 in size and 8-bit gray scale. The smallest image consisted of one image of 256×256 in size. The second images consisted of two images of 256×256 in size. The largest learning image consisted of 20 images of 256×256 in size. In this manner, we made 20 learning images and also constructed 20 code books using these learning images. Four clustering algorithms were used to construct the code books. A total of 80 code books were constructed. We selected four test images that did not include learning images (Cat512, City512, Crows512, Girlface512). These images were 512×512 pixels in size and 8-bit gray scale. Each learning image was segmented 4×4 pixels and the number of code vectors was 256 for each code book.

Fig. 3 shows changes in *PSNR* with increases in the number of learning images for the test images. In the Cat512 image, it seems that *PSNR* increases gradually as the number of learning images increases for all clustering algorithms as shown in Fig. 3. However, when the number of learning images are more than 10, *PSNR* curve becomes flat. The *PSNR* differences among clustering algorithms are small. For other curves of *PSNR*, the curves become flat when the number of learning images are more than 10. Furthermore, the *PSNR* difference among four clustering algorithms are small as well as Cat512 image.

We constructed five learning images using images shown in Fig. 2. For learning image 1, we picked up 11 images from the set of images. The images were picked up in raster scan order. In the same manner, learning image 2 consisted of 12 images and learning images 3 consisted of 13 images. Learning images 4 and 5 consisted of 14 and 15 images, respectively. Five code books were constructed using these learning images. Since there are four clustering algorithms, total 20 code books were constructed. The number of code vectors was 256 and the learning images were segmented 4×4 pixels. The same four test images were employed to examined the performance of each clustering algorithm. Table II shows the results. The minimum *PSNR* value was 28.36 in the case of the learning image 5, Crowd512 and ELBG algorithm. The maximum *PSNR* value was 30.87 in the case of the learning image 1, Girlface512, and KMC algorithm. The difference is 2.51 and this value is significant for us to perceive image quality.

However, we computed the average *PSNR* of four test images in each clustering algorithm. The average values are 29.48 for KMC, 29.37 for ELBG, 29.51 for FKM and 29.44 for FLVQ. The difference among the average values were too small (0.14) for us to perceive image between compressed image. In the end, these results indicate that selection of clustering algorithm is not important to design a code book when the learning images are different from test images. This result is the same as the result we obtained in (Sakakita, 2014).

4 CONCLUSIONS

We examined effect of clustering algorithms to design a code book for vector quantization. Four widely used clustering algorithms were selected for examination. Two sets of experiments were carried out for examination. In the first set of experiments, we examined the

Table 2: The *PSNR*s(dB) of four clustering algorithms. *PSNR*s were rounded at the second decimal places.

learning images 1 test image	KMC	ELBG	FKM	FLVQ
Cat512	29.16	29.01	28.94	28.90
City512	29.65	29.53	29.62	29.50
Crowd512	28.60	28.56	28.64	28.60
Girlface512	30.87	30.56	30.82	30.80

learning images 2 test image	KMC	ELBG	FKM	FLVQ
Cat512	29.07	29.06	29.05	28.98
City512	29.64	29.50	29.70	29.58
Crowd512	28.61	28.54	28.66	28.60
Girlface512	30.76	30.44	30.85	30.71

learning images 3 test image	KMC	ELBG	FKM	FLVQ
Cat512	29.07	29.04	29.00	29.10
City512	29.62	29.51	29.74	29.60
Crowd512	28.55	28.58	28.63	28.60
Girlface512	30.68	30.47	30.87	30.64

learning images 4 test image	KMC	ELBG	FKM	FLVQ
Cat512	29.07	29.03	29.06	29.00
City512	29.57	29.47	29.67	29.59
Crowd512	28.57	28.52	28.63	28.55
Girlface512	30.68	30.47	30.80	30.65

learning images 5 test image	KMC	ELBG	FKM	FLVQ
Cat512	28.95	28.81	28.82	28.87
City512	29.46	29.45	29.53	29.48
Crowd512	28.37	28.36	28.47	28.42
Girlface512	30.65	30.52	30.69	30.59

performance of the four clustering algorithms using the Lenna image. The Lenna image was used to construct a code book and also used as a test image. The results indicated that selection of a clustering algorithm is important. In the second set of experiments, code books using five learning images in which were different from test images. There were small performance differences among clustering algorithms. The differences were too small for us to perceive them. The results indicated that selection of a clustering algorithm is not important for constructing a code book when the learning images are different from test images.

ACKNOWLEDGEMENT

This research project is partially supported by granting-in-aid for scientific research of Japan Society for Promotion of Science (24500270).

Figure 2: 20 images to construct learning images.

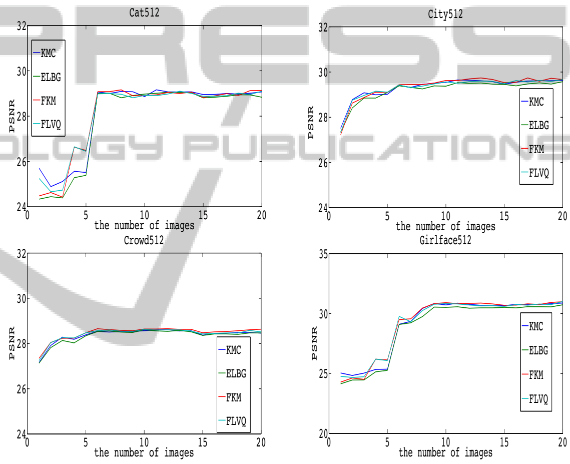

Figure 3: Changes in *PSNR*(dB) with increasing the number of learning images for the test images of Cat512, City512, Crowd512, and Girlface512.

REFERENCES

H. A. Abbass, MBO: Marriage in honey bees optimization a haplometrosis polygynous swarming approach, Proceedings of the 2001 Congress on Evolutionary Computation, 1, pp. 207-214, 2001.

C. Amerijckx, M. Verleysen, P. Thissen, J-D. Legat, Image compression by self-organizing Kohonen map, IEEE Trans. Neural Networks, 9, pp.503-507, 1998.

J. C Bezdek, Pattern recognition with fuzzy objective function algorithms, Plenum Press:New York, 1997.

H. Feng, C. Chen, F. Ye, Evolutionary fuzzy particle swarm optimization vector quantization learning scheme in image compression, Expert Systems Applications, 32, pp. 213-222, 2007.

M. Fujibayashi, T. Nozawa, T. Nakayama, K. Mochizuki, M. Konda, K. Kotani, S. Sugawara, T. Ohmi, A still-image encoder based on adaptive resolution vector quantization featuring needless calculation elimination architecture, IEEE Journal of Solid-State Circuit, 38, 5, pp. 726-733, 2003.

R. C. Gonzalez, R. E. Woods, Digital image processing, Pearson Prentice Hall:Upper Saddle River, 2008.

M-H Horng, Vector quantization using the firefly algorithm for image compression, Expert Systems with Applications, 39, pp.1078-1091, 2012.

S. M. Hosseini, A-R. Naghsh-Nilchi, Medical ultrasound image compression using contextual vector quantization, Computer in Biology and medicine, 42, pp. 743-750, 2012.

W-L Hung, D-H. Chen, M-S. Yang, Suppressed fuzzy-soft learning vector quantization for MRI segmentation, Artificial Intelligence in Medicine, 52, pp. 33-43, 2011.

A. K. Jain, M. N. Murty, P. J. Flynn, Data clustering: a review, ACM Computing Surveys, 31, 3, pp. 264-323, 1999.

N. B. Karayiannis, P. Pai, Fuzzy vector quantization algorithms and their application in image compression, IEEE Trans. Image Processing, 4, 9, pp. 1193-1201, 1995.

A. Laha, N. R. Pal, B. Chanda, Design of vector quantizer for image compression using self-organizing feature map and surface fitting, IEEE Trans. Image Processing, 13, 10, pp. 1291-1303, 2004.

S. Lazebnik, M. Raginsky, Supervised learning of quantizer codebooks by information loss minimization, IEEE Trans on PAMI, 31, 7, pp. 1294-1309, 2009.

Y. Linde, A. Buso, R. Gray, 1980. An algorithm for vector quantization design, IEEE Trans Communications, 28, 1, pp. 84-94, 1980.

H. Matsumoto, F. Kichikawa, K. Sasazaki, J. Maeda, Y. Suzuki, Image compression using vector quantization with variable block size division, IEEJ Trans EIS, 130, 8, pp. 1431-1439, 2010.

T. Miyamoto, Y. Suzuki, S. Saga, J. Maeda, Vector quantization of images using fractal dimensions, 2005 IEEE Mid-Summer Workshop on Soft Computing in Industrial Application, pp. 214-217, 2005.

B. P. Nguyen, C-K. Chui, S-H. Ong, S. Chang, An efficient compression scheme for 4-D medical imaging using hierarchical vector quantization and motion compensation, Computer and Biology and Medicine, 41, pp. 843-856, 2011.

G. Patane, M. Russo, The enhanced LBG algorithm, Neural Networks, 14, pp. 1219-1237, 2001.

H. Sakakita, H. Igarashi, J. Maeda, and Y. Suzuki, Evaluation of clustering algorithms for vector quantization in practical usage, IEEJ Trans. Electronics, Information and Systems (accepted for publication).

K. Sasazaki, S. Saga, J. Maeda, Y. Suzuki, Vector quantization of images with variable block size, Applied Soft Computing, 8, pp. 634-645, 2008.

K. Sayood, Introduction to data compression, Morgan Kaufmann Publisher:Boston, 2000.

E. C-K. Tsao, J.C. Bezdek, N.R. Pal., Fuzzy Kohonen clustering networks, Pattern Recognition, 27, 5, pp. 757-764, 1994.

G. E. Tsekouras, A fuzzy vector quantization approach to image compression, Applied Mathematics and Computation, 167, pp. 539-560, 2005.

G. E. Tsekouras, M. Antonios, C. Anagnostopoulos, D. Gavalas, D. Economou, Improved batch fuzzy learning vector quantization for image compression, Information Sciences, 178, pp. 3895-3907, 2008.

D. Tsolakis, G.E. Tsekouras, J. Tsimikas, Fuzzy vector quantization for image compression based on competitive agglomeration and a novel codeword migration strategy, Engineering applications of artificial intelligence, 25, pp. 1212-1225, 2012.

Y. Wang, X. Y. Feng, X, Y. X. Huang, D. B. Pu, W. G. Zhou, Y. C. Liang, C-G. Zhou, A novel quantum swarm evolutionary algorithm and its applications, Neurocomputing, 70, pp. 633-640, 2007.