New Adaptive Color Quantization Method Based on Self-Organizing Maps

Chip-Hong Chang, *Senior Member, IEEE*, Pengfei Xu, Rui Xiao, and Thambipillai Srikanthan

***Abstract*—Color quantization (CQ) is an image processing task popularly used to convert true color images to palletized images for limited color display devices. To minimize the contouring artifacts introduced by the reduction of colors, a new competitive learning (CL) based scheme called the frequency sensitive self-organizing maps (FS-SOMs) is proposed to optimize the color palette design for CQ. FS-SOM harmonically blends the neighborhood adaptation of the well-known self-organizing maps (SOMs) with the neuron dependent frequency sensitive learning model, the global butterfly permutation sequence for input randomization, and the reinitialization of dead neurons to harness effective utilization of neurons. The net effect is an improvement in adaptation, a well-ordered color palette, and the alleviation of underutilization problem, which is the main cause of visually perceivable artifacts of CQ. Extensive simulations have been performed to analyze and compare the learning behavior and performance of FS-SOM against other vector quantization (VQ) algorithms. The results show that the proposed FS-SOM outperforms classical CL, Linde, Buzo, and Gray (LBG), and SOM algorithms. More importantly, FS-SOM achieves its superiority in reconstruction quality and topological ordering with a much greater robustness against variations in network parameters than the current art SOM algorithm for CQ. A most significant bit (MSB) biased encoding scheme is also introduced to reduce the number of parallel processing units. By mapping the pixel values as sign-magnitude numbers and biasing the magnitudes according to their sign bits, eight lattice points in the color space are condensed into one common point density function. Consequently, the same processing element can be used to map several color clusters and the entire FS-SOM network can be substantially scaled down without severely scarifying the quality of the displayed image. The drawback of this encoding scheme is the additional storage overhead, which can be cut down by leveraging on existing encoder in an overall lossy compression scheme.**

***Index Terms*—Color image processing, color quantization (CQ), neural network, self-organizing maps (SOMs).**

I. Introduction

VECTOR QUANTIZATION (VQ) [7], [13] is one of the commonly used clustering and compression algorithms. It produces an approximation to a continuous probability density function $p(x)$ of the vectorial input variable $\boldsymbol{x}$ using a finite number of codebook vectors $\boldsymbol{m_i}$, $i = 1, 2, \ldots, k$. Once the codebook is chosen, $\boldsymbol{x}$ is approximated by finding the reference vector m_c closest to it. Based on Shannon's rate-distortion theory, coding a multidimensional ordered set of signals can achieve a superior performance compared to coding each signal component individually [6], [22]. Color quantization (CQ) [2], [17]–[21], [24], [27] is a typical method that utilizes Shannon's theory to cluster and compress color images by selecting a small number of code vectors from a universal set of available colors to represent a high color resolution image with minimum perceptual distortion. CQ is necessary for displaying continuous-tone color images on monitors that lack full color frame buffers. Even if the cost of 24-b/pixel frame buffers will become affordable for embedding into the general consumer electronics, CQ can still help to relieve the valuable frame buffer spaces for animation, transparency, window applications and other graphical functions. Besides reducing the storage requirements, CQ can also be used to reduce the transmission bandwidth of color images. Since human vision has considerable less spatial acuity for chrominance than for luminance [23], [24], in addition to the standard lossy compression that aims at removing the high-frequency components which are not visible to human vision, further compression can be achieved through CQ. A typical high-efficiency loss compression scheme for gray level images [1] can be devised to replace the VQ by CQ for color images, as shown in Fig. 1. The indices after mapping can be further compressed by the entropy encoder or other lossless compression engine already existed in such an overall lossy compression scheme. To achieve a good overall rate-distortion performance, it is important that the color quantizer possesses strong topological clustering property to preserve the neighboring pixel relationship in the mapping.

Two tasks are involved in CQ: the color palette design and quantizer mapping. Let $\boldsymbol{X} = (r, g, b)$ denote the pixel color value, where r, g, and b are the red, green, and blue intensities, respectively. The color palette design is to partition the input colors into M disjoint sets $\boldsymbol{C_s}$ $(s = 0, 1, \ldots, M-1)$ where M is the number of colors in the quantized image, which is usually limited by the display devices. For every color set $\boldsymbol{C_s}$, there is a representative color $\boldsymbol{c_s}$, which constitutes the color palette. The main issue in this step is to select the best possible set of representative colors $\boldsymbol{c_s}$ for the given image. The quantizer mapping is to define an appropriate mapping rule to associate each pixel of the image with a color from $\boldsymbol{c_s}$ to yield the highest quality image.

When VQ is implemented on neural network architecture, it is commonly accomplished with a competitive learning (CL) scheme. The computing mechanism of CL scheme is inherently parallel, which promises significant speedup and real-time processing capability when it is realized in massively parallel hardware architecture. Another advantage of CL is its online updating property. Unlike the batch algorithms such as the

Manuscript received July 21, 2002; revised August 18, 2003.

C.-H. Chang is with the Center for High Performance Embedded Systems, Nanyang Technological University, Singapore 639798. He is also with the School of Electrical and Electronic Engineering, Nanyang Technological University, Sinagpore 639798 (e-mail: echchang@ntu.edu.sg).

P. Xu, R. Xiao, and T. Srikanthan are with the Center for High Performance Embedded Systems, Nanyang Technological University, Singapore 639798.

Digital Object Identifier 10.1109/TNN.2004.836543

Linde, Buzo, and Gray (LBG) algorithm [15], CL schemes do not have to wait for all the training data to be stored in memory before processing can begin. A variety of CL schemes [3], [4], [8], [14], [16], typified by their unique learning rules, have been developed. The simplest and most widely used CL algorithms are based on the winner-take-all (WTA) [8] (or hard CL) paradigm, where adaptation is restricted to the *winner* that is the single neuron prototype best matching the input pattern. Restricting the winner to only a single neuron prototype greatly simplifies the hardware implementation and allows the network to degrade gracefully when a significant portion of its processing elements and interconnections are defective. Unfortunately, the simple CL algorithm is susceptible to the *underutilization* problem [9]. Significant efforts have been expended to eliminate the existence of *dead neurons* or lower their occurrences. By relaxing the WTA criteria, neural-gas network [16] and fuzzy CL [4] treat more than a single neuron as winners and update their prototypes by different extents accordingly. These algorithms can be classified under the winner-take-most (WTM) (or soft CL) paradigm. Though effective in eliminating the underutilization problem, WTM algorithms are too complex and expensive to be used for CQ. Under the WTA category, frequency sensitive competitive learning (FSCL) [14] desensitizes the order-dependent learning by incorporating a frequency sensitive component into the CL rules. The idea is to enhance the winning capability of the less competitive neurons, i.e., potential dead neurons so that they too, have equal opportunity to adapt. Based on their simulation results, the performance of FSCL is still slightly worse than that of the LBG algorithm. A convergence study of FSCL was later carried out in [5]. Founded on the concept of FSCL, Chen *et al.* [3] developed a multipath frequency sensitive organization (n-path FSO) method. To further alleviate the underutilization problem, two additional operations are employed, a delete operation to get rid of the less competitive neurons and the split operation to expand the frequent winners. The improvement in performance is achieved at the expense of much higher computational complexity. Kohonen's self-organizing maps (SOMs) [12] is a promising VQ technique which employs WTM strategy at the early stage of training and gradually become a WTA when its neighborhood size shrinks to zero. In addition to achieving good reconstruction quality with low distortion, it also realizes a topology preserving mapping through the tapering neighborhood updating scheme. Some researchers have applied it to the CQ problem with phenomenal success [18]–[21].

In this paper, a novel frequency sensitive self-organizing map (FS-SOM) is proposed. It inherits the neighborhood adaptation property of SOM so that greater efficiency will accrue to the coding and transmission of color images from the topology preserving codebook. In contrast to SOM, the learning rate of FS-SOM is adaptive to the winning frequency, or more precisely, the "update counter" of each neuron. Our proposed FS-SOM can be conceived as a generalization of FSCL with the notion of neighborhood adaptation, but it is much more intricate than a simple combination of FSCL and SOM. The adaptation rate and reconstruction performance of many SOM and FSCL algorithms are very sensitive to the setting of the initial state and the network parameters. Quality performance with acceptable training time is often accomplished through scrupulous tuning and customized tailoring of a set of network parameters. One unique feature of our FS-SOM is its robustness against key network parameter variations. The training algorithm has been designed with a holistic consideration of convergent efficiency, computational complexity in parallel implementation and performance stability. The CQ performances of our FS-SOM are evaluated and compared with several FSCL and recent SOM methods optimized for this application. To increase the number of colors and, hence, the perceptual quality of the reconstructed images without expanding the size of the network, we also propose a most significant bit (MSB) biased encoding scheme. The overhead of additional storage required by this method is minimized by leveraging on the entropy encoder existing in the overall lossy compression scheme shown in Fig. 1. This scheme is particularly attractive for hardware implementation of SOM due to the conflict between the prohibitively costly VLSI area to realize a large number of processing elements and the desirable high visual quality of the reconstructed images. Computer simulation shows that the MSB-biased encoding has contributed to either an exceedingly high-quality output or a significant reduction of neurons for a given performance. Thus, the proposed enhancement technique provides the added flexibility to mitigate system cost for an acceptable reconstructed output quality.

The rest of the paper is organized as follows. In Section II, the basic structure of SOM and its fundamental training algorithm for the CQ application are described. Section III presents the underlying principles and evolution of the proposed FS-SOM algorithm and MSB-biased encoding technique. In Section IV, analysis and discussion on the simulation results of our proposed algorithms and the comparison against other related methods are provided. The paper is closed with a conclusion in Section V.

II. SOMs

A typical Kohonen network consists of two layers, an input layer and an output layer. The input layer operates as a buffer for the input vectors $\boldsymbol{X} = [x_1, x_2, \ldots, x_n]^T \in \Re^n$, whereby the inputs are transmitted parallelly to all the neurons in the output layer. In the output layer, which is also called the competitive layer, the neurons are interconnected to each other to form a linear, planar or multidimensional structured topology. Every neuron i is associated with a parametric reference vector $\boldsymbol{w_i} = [w_{i1}, w_{i2}, \ldots, w_{in}]^T \in \Re^n$. Fig. 2 shows a typical SOM structure in which the input data space $\Re^n$ is mapped onto a two-dimensional (2-D) array of neurons [12].

As indicated in Fig. 2, all neurons are presented with the same input vector $\boldsymbol{x}$ in parallel, and all neurons compute its distances between their weights $\boldsymbol{w}$ and the input vector $\boldsymbol{x}$ in parallel. Only the neuron with the closest match between the input vector and its weight produces an active output. Each neuron therefore acts as a separate decoder (or pattern detector, feature detector) for the same input and the interpretation of the input is derived from the presence or absence of an active response at each neuron. The lateral interaction between neurons introduces a sense of topology, such that neighboring units represent inputs that are close together in the input space [12]. Therefore, SOM is also referred to as topology-preserving maps.

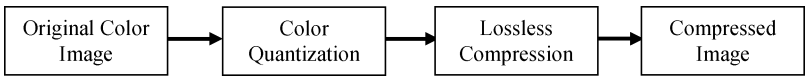


Fig. 1. Structure of an overall lossy color image compression scheme.

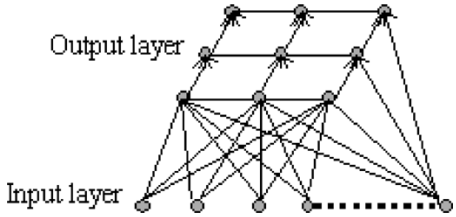


Fig. 2. Structure of a 2-D SOM network.

When the SOM network with N neurons is used to perform CQ, three main phases of operations are involved: the initial network setup (*Step 1*), the neurons training to obtain the color palette (*Steps 2–5*) and the quantizer mapping to generate the pixel indices (*Step 6*). Let N be the total number of neurons and $\boldsymbol{w_i}(t) \in \Re^3$ be the weight vector of the ith neuron at epoch time t. Let $h(t)$, $\alpha(t)$ and $\beta(t)$ denote the neighborhood size, the learning rate for the winning neuron and the learning rate for the neighborhood of the winning neuron, respectively.

Step 1) At $t = 0$, the weights of the neurons, $\boldsymbol{w_i}(0) = [w_{ir}, w_{ib}, w_{ig}]^T$, $i = 0, 1, \ldots, N-1$, are initialized according to a selected initialization criterion. The network topology is defined and the size of the neighborhood, $h(0)$ is initialized. The initial learning rates, $\alpha(0)$ for the winner and $\beta(0)$ for neighbors of the winning are also defined.

Step 2) A new image pixel value $\boldsymbol{x}(t) = [r(t), g(t), b(t)]^T$ is fetched according to a predefined pixel ordering scheme. t is set to 0 for the first input pixel.

Step 3) The training process is conducted in the competitive layer. The input tristimulus $\boldsymbol{x}(t)$ is compared to the weight $\boldsymbol{w_i}(t)$ of each neuron i simultaneously, by the square of Euclidean distance in red, green, and blue (RGB) space defined as follows:

$$\|\boldsymbol{x}(t) - \boldsymbol{w}_i(t)\|^2 = (r(t) - w_{ir}(t))^2 + (g(t) + w_{ig}(t))^2 + (b(t) - w_{ib}(t))^2. \quad (1)$$

The winning neuron c for this input is the one with the smallest distance, i.e,

$$c = \arg\min_i \|\boldsymbol{x}(t) - \boldsymbol{w}_i(t)\|^2. \quad (2)$$

The winning neuron, c and all the neurons belonging to the neighborhood of c, denoted as b, are trained from the input vector $\boldsymbol{x}(t)$. The update is done by biasing the weights of the neurons as follows:

$$\boldsymbol{w}_c(t+1) = \boldsymbol{w}_c(t) + \alpha(t) \times [\boldsymbol{x}(t) - \boldsymbol{w}_c(t)] \quad (3)$$

$$\boldsymbol{w}_b(t+1) = \boldsymbol{w}_b(t) + \beta(t) \times [\boldsymbol{x}(t) - \boldsymbol{w}_b(t)] \quad (4)$$

where b is the neighborhood neurons of c defined by a specific neighborhood function.

Step 4) The epoch time t is incremented by one. The learning rates are reduced according to a linear or an exponential function of t. The neighborhood size $h(t)$ is also shrunk monotonically in time. The process is repeated from Step 2 until the average difference in the neuron weights between two successive iterations converges to a small value.

Step 5) The colormap design has been completed. The weights $\boldsymbol{w_i}$ of the neurons constitute the colormap of the encoded image.

Step 6) For every pixel value in the original color image, the best representative color is found from the colormap based on the same Euclidean distance metric of (1).

In short, the basic SOM algorithm consists of drawing sample vectors from the input pixels and "teaching" them to the neurons in the output layer. The teaching is done by choosing a best matching unit by means of a similarity measure and by updating the weights of the neurons in the neighborhood of the best matching unit. This process is repeated a number of times. In this manner, the net tends to approximate the probability density function of the input data. The learning rates, $\alpha(t)$ and $\beta(t)$ are used to control the amount of bias added to the winning neuron and its neighborhood. To emulate a fast learning, slow forgetting training process, the learning rates are set high initially. As more input pixels are presented to the network, both $\alpha(t)$ and $\beta(t)$ are gradually reduced to zero according to a linear or an exponential function. With large initial learning rate, the network is able to adapt faster to the clusters of input data at the beginning of the training stage. The neuron weights are fine tuned at the later stage with decreasing $\alpha(t)$ to guarantee convergence. However, this basic solution is effective only when each neuron has an equal probability to win and update throughout the entire training process. In practice, colors in an image are generally not uniformly distributed. Without special calibration and optimization, when the neurons finally settled down, the weights of the network may not be optimally placed in the input space.

III. FS-SOM

A. FS-SOM

FSCL is claimed to be effective in eliminating the underutilization problem and can be easily implemented in hardware [14]. However, it lacks the topology preserving property of SOM, which is very useful in coding and transmission. For example, in [20], the SOM topology is explored to further compress the encoded image to increase its coding efficiency. Although SOMs have been successfully used in CQ of still and video images [20], [21], its performance is often dependent on the optimal selection of training parameters presented in the previous section. Sometimes a small change in the setting of the SOM can result in a significant variation on its performance. Once the network is calibrated for a particular source of inputs, it is very difficult for the end user to readjust the SOM

parameters to achieve the same optimal performance under an alternative source. It is as important and preferable for the training algorithm to yield stable performance under different settings of network parameters. Besides circumventing the underutilization problem, the proposed FS-SOM is also designed to sustain a robust performance. In the sequel, the details of each step of the FS-SOM algorithm will be described with a 1-D string topology.

1) Initialization of FS-SOM: Because of the topological neighborhood of FS-SOM, initialization of neuron weights is not as crucial as for some other clustering algorithms, like LBG. A better initialization scheme will enable the network to converge faster but it requires a priori knowledge of the input data distribution. If nothing or little is known about the input data at the time of initialization, we initialize the neurons of FS-SOM with uniformly distributed weight vectors on the major diagonal of the input space, i.e

$$\boldsymbol{w}_i = \left[\left(i-\frac{1}{2}\right)\times\frac{255}{N},\ldots,\left(i-\frac{1}{2}\right)\times\frac{255}{N}\right]^{\mathrm{T}} \quad \forall i=1,2,\ldots,N \tag{5}$$

where N is the number of neurons. This initialization method is very simple, nevertheless, it provides a better initial topological order of FS-SOM to facilitate faster convergence than the random initialization.

2) Butterfly Permutation for Input Randomization: Online learning causes the learning performance to be order dependent, particularly when the training set contains a high-degree of redundant information. Therefore, training the FS-SOM with a raster scanning order of the input pixels has a strong tendency to over-train some groups of neurons. Here, a global butterfly permutation sequence is proposed to present the spatially correlated input data from a multidimensional coordinate system. The aim is to let the neurons learn the characteristics of the training source as early as possible to prevent the performance from being degraded by the order dependent learning. The proposed global butterfly permutation is a generalization of the block-based butterfly jumping sequence in [20] whereby the pixels are sub-sampled from each block to form different training sweeps. It is defined by a mapping, $\pi : Z \rightarrow Z^n$ of an input order number $r \in Z$ to a n-dimensional coordinate system, $(x_1, x_2, \ldots, x_n) \in Z^n$ where Z is a finite integer space. For $x_i \in [0, N)$, the butterfly permutation π can be expressed mathematically as

$$\begin{aligned}\pi(r) = \Bigg\{ & (x_1, x_2, \cdots, x_n) \in Z^n \big| x_j \\ & = \sum_{i=0}^{\lceil \log_2 N\rceil - 1} 2^{\lceil \log_2 N\rceil - 1 - i}(r_{ni} \oplus r_{ni+j}) \forall j \neq n, x_n \\ & = \sum_{i=0}^{\lceil \log_2 N\rceil - 1} 2^{\lceil \log_2 N\rceil - 1 - i} r_{ni} \Bigg\}\end{aligned} \tag{6}$$

where r_i is the ith bit of the binary representation of the decimal number r, and r_0 is the least significant bit.

3) Winning Neuron Search: To allow every neuron to compete fairly and learn adequately irrespective of its winning sequence, a modified distortion metric, $D_i = F(u_i)\|\boldsymbol{x_i}(t) - \boldsymbol{w_i}(t)\|^2$ is defined in FSCL [14] where an update counter, u_i is used by each neuron to keep track of the number of times it wins. The function, $F(u_i)$ is designed to improve the competitiveness of the less frequently updated neurons. As our FS-SOM has incorporated several effective measures like neighborhood adaptation, frequency sensitive update, and dead neuron reinitialization to overcome the underutilization problem, we can avoid the difficulty of designing an optimal $F(u_i)$ by adopting the standard SOM best marching unit search with $F(u_i) = 1$.

4) Updating of Winning Neurons and Their Neighborhoods: The weight vectors and the update counters of the winning neuron and its neighborhood are adjusted according to the following equations:

$$w_i(t+1) = w_i(t) + G(u_c)H(i,c,m)\,[x(t) - w_i(t)] \tag{7}$$

$$u_i(t+1) = u_i(t) + H(i,c,m) \tag{8}$$

where $G(u_c)$ is the frequency sensitive learning rate function. It is a monotonic decreasing function of u_c, and its value is usually bounded in the range (0,1). $H(i,c,m)$ is a tapering neighborhood function which is a hallmark of the SOM algorithms. This neighborhood taper alters the update of each neuron, i within the neighborhood of the winning neuron, c by a different amount at discrete time interval, m depending on the distance between the weight vectors w_i and w_c. It is this neighborhood taper that gives SOM its topological ordering property. The following neighborhood taper is used:

$$H(i,c,m) = e^{-\frac{\|w_i - w_c\|^2}{\sigma(m)^2}} \tag{9}$$

$$\sigma(m) = N\beta k^m \tag{10}$$

where m represents the number of sweeps. A sweep is a period during which the updating parameters remain constant. In our simulations, we set the number of input data in one sweep to be 12 times the number of neurons, i.e.

$$m = \left\lfloor \frac{t}{12N} \right\rfloor. \tag{11}$$

On average, each neuron will be updated 12 times as the winning neuron in a sweep. $\sigma(m)$ is the neighborhood width at the sweep number m. It defines the boundary of the lateral interaction between the winning neuron c and other neurons. β defines the extent of the initial neighborhood interaction. It is usually set to a value between 0 and 0.5, where $\beta = 0.5$ implies that the neighborhood interaction exists in the entire network initially, and $\beta = 0$ implies that there is no neighborhood interaction at all. The constant, $k < 1$ and it is used to control the rate at which the boundary of the neighborhood shrinks.

In (7) and (8), the use of a frequency sensitive learning rate function implies that the training time for each individual neuron, rather than the training time of the network, is used to modulate the distance metric for updating. As this mode of learning uniquely characterizes our FS-SOM from other SOM algorithms, it is important to establish a "fast adaptation and slow forgetting" learning function to ensure that the weights settle well during training. We experiment with the nonlinear reciprocal function, $G(u_c) = u_c^{-l}$ for different values of $l > 0$.

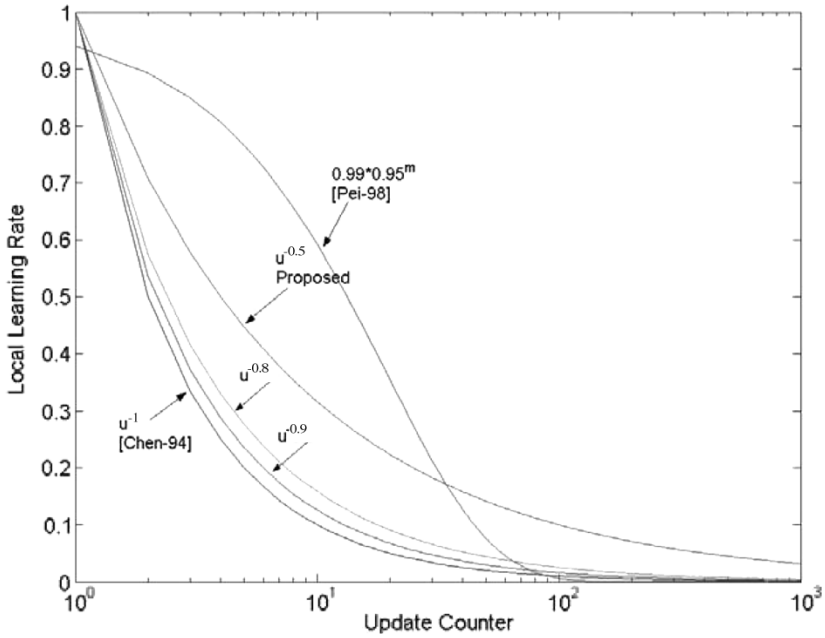


Fig. 3. Comparison of different learning rate functions.

It is found that $l = 0.5$ gives excellent results for different test images and it is set as the default value for our simulation. The comparison of the proposed learning rate function and some popular learning rate functions [3], [20] used by other researchers is shown in Fig. 3.

It is noticed that the proposed learning rates strive to drag the winning neuron toward the input cluster as quickly as possible within its first 10 to 50 updates, thereafter the neuron is allowed to fine tune gradually in its later wins. For other learning rate functions, the neurons are almost frozen after 50 times of winning.

The intrinsic learning mechanism of our proposed FS-SOM algorithm is fundamentally different from the training algorithm proposed by Pei and Lo [20]. The proposed learning rate function $G(u_c)$ is dependent on the update counter value of the winning neuron. Hence, although the neurons compete globally across the network, training is adapted locally to each neuron. Whereas for the SOM algorithms, such as that of [20], the learning rate function is time dependent. As a consequence of the "globally" applied monotonic decreasing learning rate function, some of the neurons may be over trained at the beginning of the training process. The net result is that when the training process is completed, the late winners may become underutilized in the pixel mapping process. By measuring the mean Euclidean distance between the weight vectors of SOM from one time step to another at discrete interval during the training period, it is observed that the choice of $H(i, c, m)$ has a significant influence on the performance stability of SOM. In contrary, supplemented by the frequency sensitive learning, the proposed FS-SOM is less sensitive to poorly calibrated network parameters in view of its stability in VQ performance. Even with the simplistic uniform or linear neighborhood function, the frequency sensitive learning scheme will still manage to resolve the underutilization problem to a large degree. The robustness of VQ performance of our FS-SOM over various functions of $H(i, c, m)$ will be demonstrated in the simulation results to be presented later. However, a suitable neighborhood function can still accelerate the topological ordering of FS-SOM as in SOM. For this reason, a Gaussian neighborhood taper of $H(i, c, m)$ used by [20] has been selected.

5) Convergence Criteria: At the end of each sweep, the weight vectors of the neurons in the current and the previous sweeps are compared. If their difference is less than a predefined small threshold value, then the training is completed. The convergence criterion can be stated as

$$\frac{\sum_{i=1}^{N} \|w_i(m) - w_i(m-1)\|}{N} \leq \varepsilon. \tag{12}$$

In our simulation, $\varepsilon = 0.01$. In other words, if the average square difference of the neuron weights is less than 0.01, the training is terminated.

To further increase the stability of the algorithm, we will look for dead neurons $(u_i = 0)$ at the end of each predefined training interval. If dead neurons exist, they will be reinitialized. Otherwise, the learning process will be continued without further checking for dead neuron. By reinitialization, the weight of the most frequently updated neuron is copied to the first dead neuron. The weight of the second dead neuron will be reassigned the weight of the second most frequently updated neuron, and so on. The update counters of both neurons in each pair of neurons (one dead neuron and one corresponding frequently updated neurons) will be reset to half of that of its corresponding frequently updated neuron in each pair. If the network parameters are well selected, this reinitialization operation will never be needed. However, if one or more of the parameters cause SOM to adapt poorly, the existence of the dead neurons will quickly be detected and corrected by this reinitialization process. In [3], a similar concept but different approach is adopted. The major difference is that: in [3], after checking and reinitializing the

dead and inactive neurons, the update counters of all neurons are refreshed to zero and a new learning cycle is started. In our proposed algorithm, we halve the update counter and continue with the learning instead of resetting them to zero and starting a new learning cycle. In this way the detection and reinitialization is more purposeful, faster and effective. Furthermore, we only detect and reinitialize the neurons with zero update rather than all neurons with low winning frequency. This is because the frequency sensitive CL rate function has allowed the less competitive neurons to learn adequately. Only the dead neurons, which have no opportunity to update at all, need to be dealt with. Another reason is that, the minority color can be very important in some CQ applications. For example, in the images of roadmaps or landscapes, it is useful to preserve the minority colors of prominent landmarks with less utilized but sufficiently trained neurons. It is unwise to sacrifice them completely to trade for a slightly lower global distortion.

B. MSB-Biased FS-SOM

When SOM is used for CQ, the color palette is formed from the final weights $\boldsymbol{w_i}$ of the neurons. Hence, the maximum number of colors that can appear in the reconstructed image is equal to the number of neurons, N. As a result, large size neural network is required to eliminate the commonly encountered visual artifacts due to limited available colors, such as color shift and false contouring [17], [20], [24], which is a phenomenon where regions of slow color variation is mapped with abrupt color transition across region boundaries. Currently the only way to increase the color resolutions for the CL based methods is to increase the number of neurons.

Taking into account of the overall effect of visual quality, number of colors and the cost factor of the neural network, we propose a change in the basic number representation of the input tri-stimulus values to the network, so that more colors can still be represented with the same number of neurons to achieve better perceptual quality.

The idea is to collapse the original $256 \times 256 \times 256$ RGB cube to a denser octa-cube of size $128 \times 128 \times 128$ by mapping eight scattering clusters to a point density function within the octa-cube, as shown in Fig. 4. If the mapping is reversible, the full range of 16 million colors in the original cube can still be recovered when the octa-cube is expanded. Consequently FS-SOM CQ is performed in this octa-cube where the red, green and blue components of the neurons are each scaled down to 7 b. In other words, each neuron is now capable of representing eight different color clusters of the image in the original RGB cube. To reduce the spatial scale, a coding scheme has to be applied to the RGB components (8 b each) of each input pixel. Here we use a MSB biased method.

In MSB-biased method, a coding scheme resembling the excess biased number representation is applied to each RGB component of the input pixels. Each R, G, and B color component [0, 255] is biased by subtracting from it a constant value of 128 to obtain a sign-magnitude number in the range of -128 to 127. Only the magnitude of the number (7 b) is presented to FS-SOM for training. To avoid the asymmetrical maximum positive and negative magnitudes, the negative magnitude is further decremented by one. This clever approximation leads to a very simple

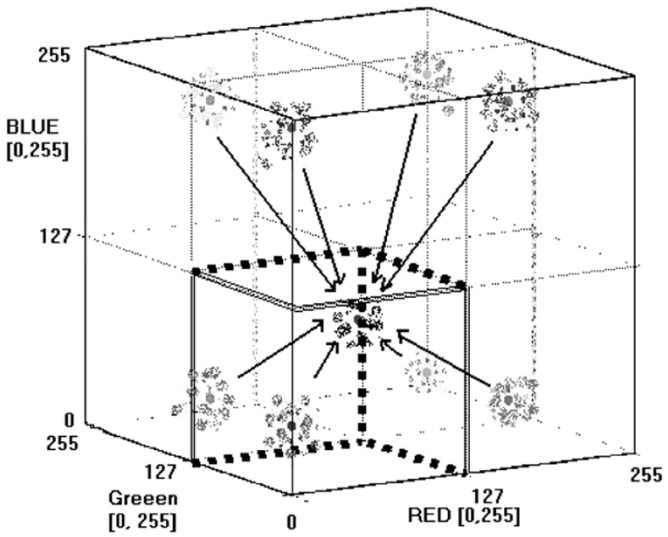


Fig. 4. Spatial reduction mapping of RGB cube.

method of encoding the input pixels in hardware implementation. That is for each color component x, if the MSB, $x7$ is "0," the 7-b magnitude is obtained from the 1's complement of its seven lower order bits, $x6\ x5\ \ldots x0$; If $x7$ is "1," the seven lower order bits are taken directly as the magnitude, i.e.

$$\text{if } (x7 = 1)\ X = X(6 \ldots 0)$$
$$\text{else } X = \sim X(6 \ldots 0).$$

The MSB planes for each input pixel are stored for the reverse mapping. Since FS-SOM preserves the topology between the input and output spaces, it is important for the seven lower order bits that fed into FS-SOM to maintain approximately the same intensity differential of two spatially adjacent pixels in the input space. The selective complement serves this purpose.

The complete flow chart for the MSB-biased CQ method is shown in Fig. 5. In the decompression process, with the aid of the MSB planes, the seven lower order bits of each color component are recovered and appended to the corresponding MSB to reconstruct the quantized image.

The storage overhead is the first-bit plane of each color channel, which is 3 b per pixel. The recent bilevel lossless compression techniques, such as the well-known bilevel image coding standard JBIG (ISO/IEC 11 544) and emerging JBIG2 [10], can be used to effectively compress these bits. We have verified in simulation that the excess 3-b planes can be easily compressed by JBIG to a bitstream of approximately 0.32 b per pixel. We have also used the competitive CALIC lossless encoder [26] to compress the indices of the encoded images with topological palette generated by a 256-neuron FS-SOM or SOM. The obtained bitstream is about 4.90 b per pixel. The overhead incurred by the FS-SOM with MSB-biased encoding is on average 6.5% of that without it. From the overall lossy compression scheme of Fig. 1, the indices of the pixels after CQ are usually differentially encoded before further compressed by a lossless compression engine. No additional hardware is required as the lossless encoder is there whether or not the MSB-biased scheme is used. With the use of MSB-biased encoding, however, the number of processing elements in

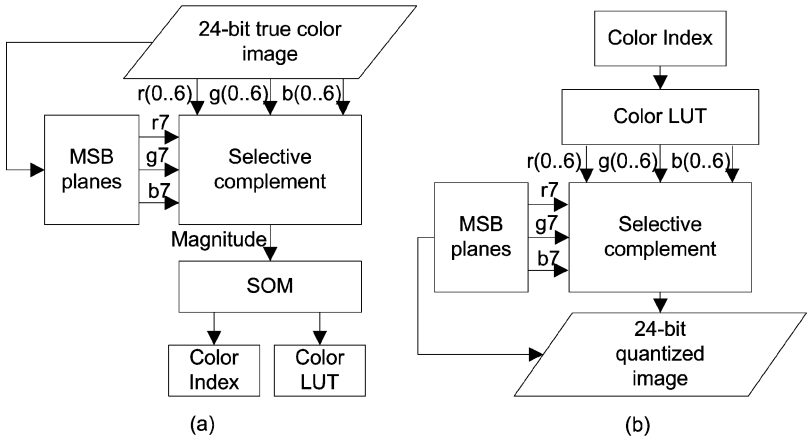


Fig. 5. MSB-biased SOM. (a) Compression. (b) Decompression.

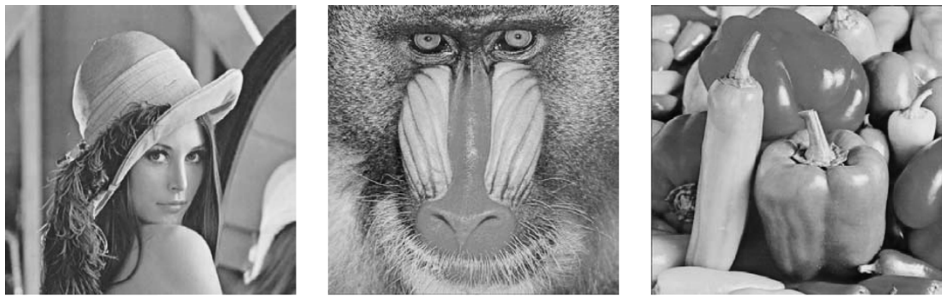


Fig. 6. Test images. (a) Lena. (b) Baboon. (c) Pepper.

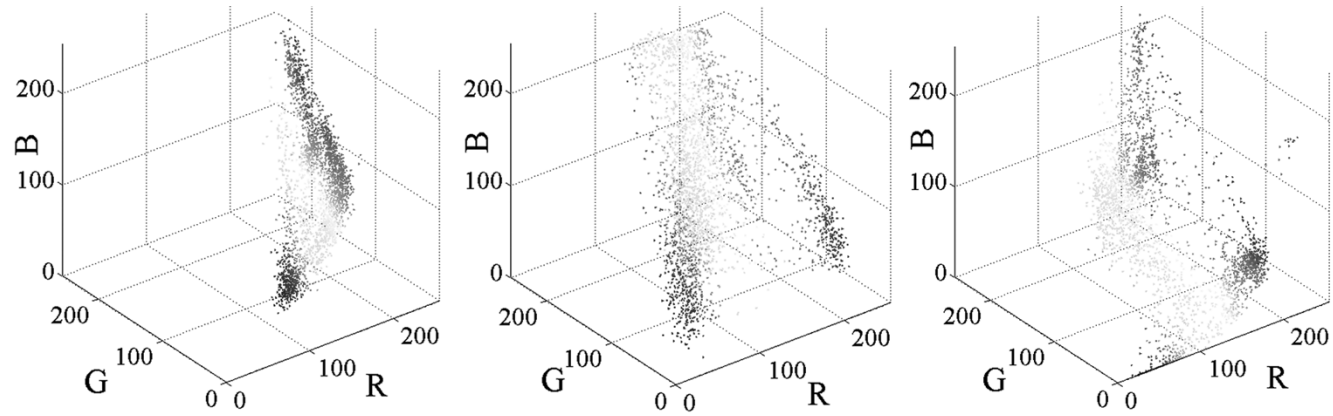


Fig. 7. Color distributions. (a) Lena (b) Baboon and (c) Pepper.

FS-SOM can be reduced significantly while maintaining the same minimal visual distortions between the quantized images and the original ones, as proven by the experimental results in the next section.

IV. Simulation Results and Discussions

For the purpose of comparison, the original 24 b per pixel true color images of Lena, Baboon and Pepper as shown in Fig. 6 are used for the simulation. Lena is a natural image with smooth color transitions. Baboon is an image demarcated by rich high-frequency texture with a much broader color range. Pepper has a profusion of small color variations and the illumination on the object surfaces has created a smooth dispersion of colors. The color distributions of these images in the RGB space are shown in Fig. 7.

Fig. 8 shows the 1-D topological maps of the final adaptation results obtained from a 64-neuron FS-SOM for the three different color images. Neighboring neurons linked by solid lines represent adjacent entries in the color palette. The absence of long links indicates a good color-ordering palette, which is a good attribute for speeding up the encoding (pixel mapping) process by local search. By superimposing the topological maps of Fig. 8 on the color distribution of the original images in Fig. 7, it is observed that more neurons are assigned to the denser color regions and lesser neurons are scattered wider and evenly over

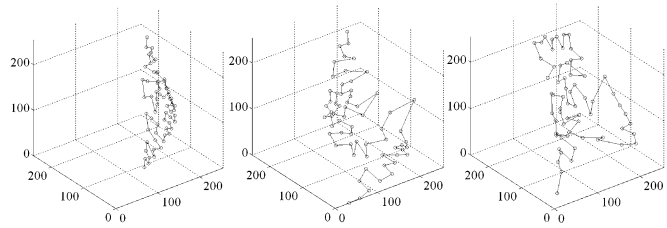


Fig. 8. Resultant 1-D topological maps of a 64-neuron FS-SOM. (a) Lena; (b) Baboon; (c) Pepper.

Fig. 9. Reconstructed images for (a) Lena. (b) Pepper. (c) Baboon under different codebook sizes: (from top to bottom) 16, 32, 64, and 128 colors.

the sparse color regions. This implies that the color palette generated by FS-SOM is able to match the pixel color distribution very well.

To assess the perceptual quality of the color images reconstructed from the color palettes generated by FS-SOM, the images of Lena, Pepper and Baboon palettized with different codebook sizes are shown in Fig. 9. No dithering operation [24] has been applied on these images in order to evaluate the real performance of FS-SOM in clustering the data to match each separate color. Even without post-processing by dithering operation, there is hardy any observable false contouring artifacts for all the reconstructed images generated from the trained FS-SOM of 64 neurons and above. Comparing with the original full color images of Fig. 6, for the 16- and 32-color images, minor contouring and false edge artifacts can be observed on the rim of the hat and arm of Lena, the cheek of Baboon and the surfaces of the large green and red peppers due to the limited number of colors. Interestingly, the perceivable artifacts appear to be less annoying in the high-texture Baboon image than the other two smooth images despite it

has a denser and wider color distribution (see Fig. 6). The less noticeable contouring artifacts may be due to the fewer and smaller regions of smooth color transitions of Baboon.

We can observe the adaptation behavior of the weights at various discrete points by measuring the mean Euclidean distance, $\delta(m)$ between the weight vectors of FS-SOM calculated at successive sweeps during the training process, i.e.

$$\delta(m) = \frac{\sum_{i=1}^{N} \|w_i(m) - w_i(m-1)\|}{N}. \tag{13}$$

The mean change in Euclidean distance $\delta(m)$ plotted against the sweep number is shown in Fig. 10 for different number of neurons. The solid lines represent the $\delta(m)$ responses of FS-SOMs with different 1-D string configuration network sizes ranging from 16 to 256 neurons, and the result of the 16-neuron 1-D string SOM of [20] is plotted as dotted line for comparison. This figure demonstrates the difference between the learning philosophies of SOM and FS-SOM. In SOM, the mean change in Euclidean distance $\delta(m)$ reduces with time gradually and steadily under an optimally tuned learning rate function which is time-decreasing and neuron-independent. Consequently, the longer the elapse time, the smaller the aggregate change in neuron weights between successive sweeps. Depending on the descending rate of the learning function, such brute force convergence (learning rate diminishes to zero as time elapses) does not necessarily guarantee that all neurons are adequately trained upon convergence. On the other hand, the learning rate function of our FS-SOM is dependent on the winning neuron's update counter. The learning rate is large when a less frequently updated neuron wins at the time of observation and the learning rate is small when a more competitive neuron is being updated. The surges on the curves correspond to the updates of infrequently win neurons. The $\delta(m)$ responses undergo three different phases. During the initial fast adaptation phase, the competitive neurons corresponding to majority colors or denser clusters are rapidly updated and settled. During the equalization phase, the skimpily updated neurons correspond to minority colors or sparse clusters are given opportunity to order themselves, giving rises to a number of spikes. After which FS-SOM enters into a final settlement phase where the response of $\delta(m)$ dips steeply till convergence. This equalization phase and the occasional drastic fluctuations of $\delta(m)$ response clearly marks the unique neuron-dependent learning behavior of FS-SOM.

The effect of reinitialization is analyzed in Fig. 11. Two situations of underutilization problem are created by training the smooth images of Lena and Pepper with poor learning parameters. The peak signal-to-noise ratio (PSNR) of the reconstructed image is used to measure the quality of the reconstruction during the training process. The PSNR is defined as follows:

$$\text{PSNR} = 10 \times \log\left(\frac{3 \times 255^2}{\text{MSE}}\right) \tag{14}$$

where $MSE = (\sum_{j=0}^{N_t-1}(\boldsymbol{X_j} - \boldsymbol{X'_j})^2)/N_t$. $\boldsymbol{X_j}$ and $\boldsymbol{X'_j}$ are the pixel values of the original and the reconstructed images, respectively. N_t is the total number of pixels.

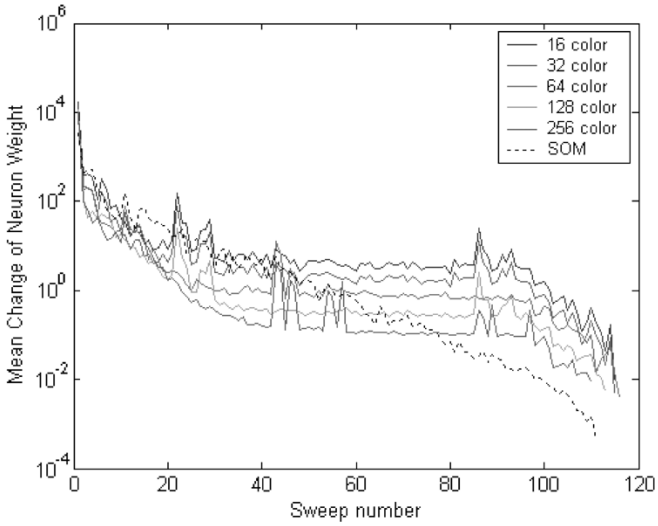


Fig. 10. Mean change of the neuron weights during the training process.

In Fig. 11, the trends of reconstruction performance during training for FS-SOM with and without reinitialization are marked, respectively, by the solid and dotted lines. Reinitialization boosts the PSNR performance of FS-SOM significantly above that without the reinitialization. As shown in Fig. 11(a), although the training of FS-SOM without reinitialization converges earlier, its final PSNR is far worse. Reinitialization assists FS-SOM to adapt the dead neurons proactively during the fast ordering phase and leave the fine adaptation of the neurons to the frequency sensitive learning scheme and neighborhood adaptation, so that when the training finally converges, a good reconstruction quality can be achieved.

The performance of the two proposed methods, FS-SOM and MSB-biased FS-SOM (MB-FS-SOM) are evaluated and compared with the SOM algorithm proposed in [20], [21] by examining the quality of the reconstructed images. The classical CL algorithm and the celebrated batch algorithm of LBG are also implemented and tested. The convergence criteria for different algorithms are kept the same. All the simulations were run on a Personal Computer equipped with a Pentium Pro. 2.1 GHz Processor and 512 MBytes of system memory. The results are shown in Table I. 1-D string topology has been used for both SOM and FS-SOM. As noted in [20], the performance differences of calibrated 1-D string and 2-D mesh structures are trivial.

From Table I, it can be seen that the basic CL scheme gives the worst performance due to its severe underutilization problem. LBG delivers much better performance than CL. However, being a batch algorithm, LBG needs the entire training set to be presented before performing an update (though some data squashing and reliable sampling techniques can be used to alleviate this problem) and has long computation time. In comparison with the SOM and proposed FS-SOM algorithms, LBG is more sensitive to initialization. The remaining algorithms all have the advantages of fast convergence and online learning capability. Despite the already superb quality of reconstruction offered by the current art SOM algorithm in CQ [20], [21], our proposed FS-SOM algorithm can still outperform it slightly for all the different color palette sizes and images

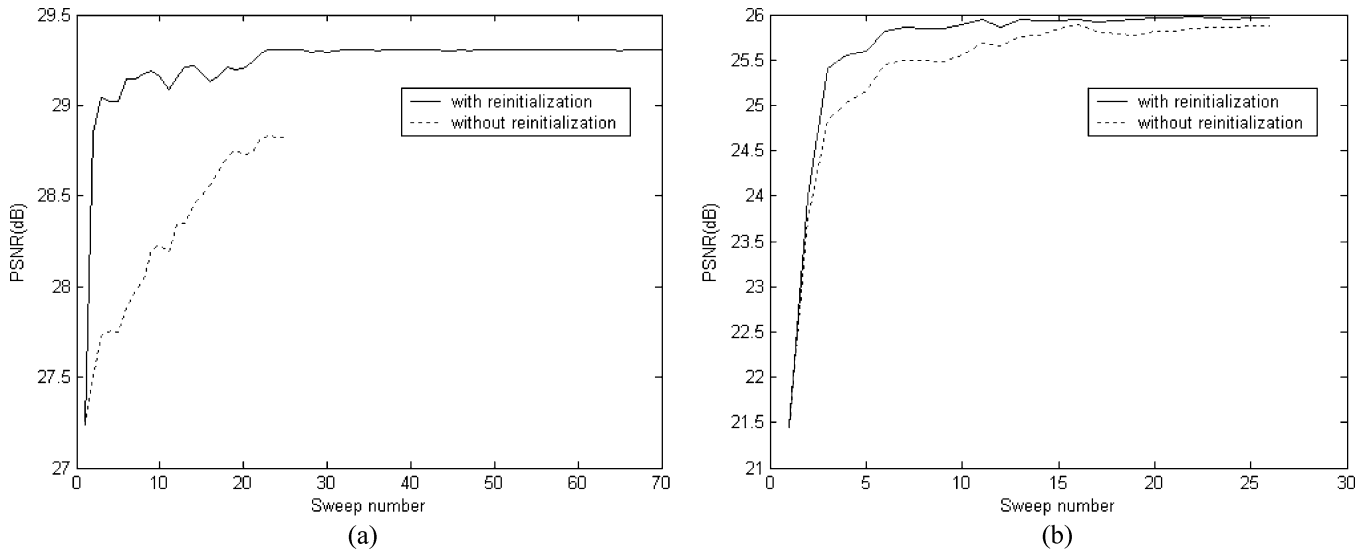


Fig. 11. Learning tracks of 16-neuron FS-SOM with and without reinitialization. (a) Lena. (b) Pepper.

TABLE I
PSNR (IN dB) OF VARIOUS ALGORITHMS WITH DIFFERENT CODEBOOK SIZES

Size of Codebook	Color Image	CL	LBG	SOM[20]	FS-SOM	MSB-biased FS-SOM
16	Lena	28.80	29.65	29.61	29.67	30.73
	Baboon	24.78	24.91	24.85	24.89	27.70
	Pepper	26.80	26.74	26.70	26.70	28.91
32	Lena	30.71	32.14	32.09	32.12	33.16
	Baboon	26.97	27.16	27.05	27.11	29.90
	Pepper	28.96	29.24	29.10	29.22	31.20
64	Lena	32.53	34.25	34.23	34.25	35.21
	Baboon	28.89	29.16	29.01	29.10	31.89
	Pepper	31.25	31.26	31.40	31.59	33.23
128	Lena	33.94	36.08	36.08	36.15	37.14
	Baboon	30.79	31.08	30.91	31.04	33.87
	Pepper	32.79	33.49	33.48	33.65	35.16
256	Lena	35.26	37.78	37.74	37.97	39.00
	Baboon	32.61	33.00	32.66	32.96	35.81
	Pepper	34.13	35.40	35.14	35.44	37.06

simulated. With the MSB-biased encoding scheme, even better reconstruction quality can be achieved by the MB-FS-SOM algorithm. The reconstructed image of MB-FS-SOM carries as much as 8 times more colors than the number of colors available by the color palette. The visual qualities of the final adaptation results obtained from the 16-neuron FS-SOM and MB-FS-SOM are compared in Fig. 12. The drawback of the decoding and storage overhead of the MSB-biased encoding scheme has been discussed in Section III.

Table II lists the training time of the different algorithms. From Table II, it is observed that the computation time of the proposed algorithms are compatible to those of the CL schemes, including the basic CL and SOM. The classical LBG is much more time consuming due to its batch updating nature. Some techniques [11], [25] can be used to speed up the LBG algorithm by intelligently reducing the number of unnecessary distance calculations. However, many of these techniques are equally applicable to the SOM and FS-SOM algorithms, giving similar or better acceleration than when they are applied to LBG. For this reason, the comparison is done between the classical LBG and other algorithms without applying the reduction techniques.

One distinct advantage of our FS-SOM algorithm is its performance robustness. As the performances of SOM and other quantization algorithms reach the margins of diminishing return, they become highly susceptible to the variation of parameters and input data distribution. Besides the parameters that control the learning rate function which have already been discussed in Section III, two other most important parameters common to the FS-SOM and SOM [20] training algorithms are β and k in (10). These parameters determine the neighborhood interaction $\sigma(\mathrm{m})$ and they need to be empirically tuned to achieve optimal quantizer performance. The performance variations over a reasonable range of these two parameters are studied for FS-SOM and SOM. The simulation results are compared in Fig. 13. The test image is Lena, and the size of the network is 16.

Fig. 13 shows that the performance of SOM fluctuates apparently with the training parameters. In the best case, it can

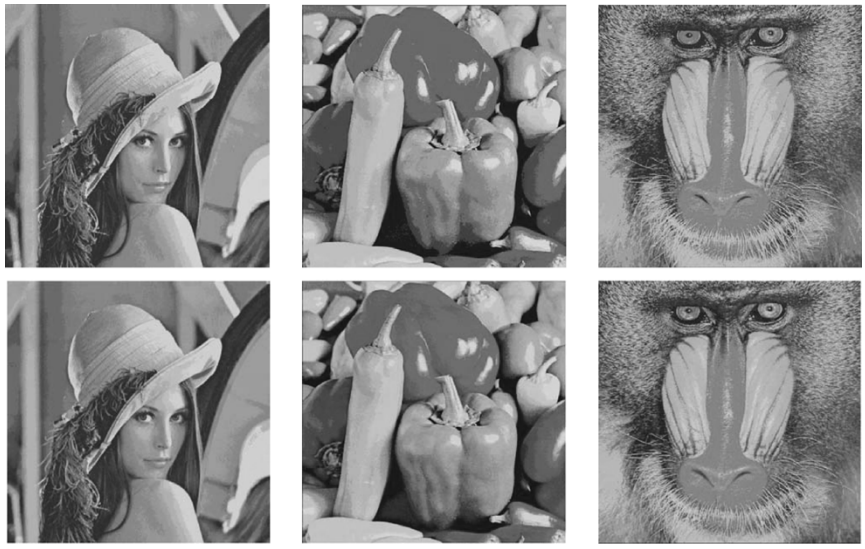


Fig. 12. Reconstructed images using 16-neuron FS-SOM (top) and 16-neuron MB-FS-SOM (bottom) for three different color images: (a) Lena. (b) Pepper. (c) Baboon.

TABLE II
TRAINING TIME (IN SECONDS) OF VARIOUS ALGORITHMS WITH DIFFERENT CODEBOOK SIZES

Size of Codebook	Color Image	CL	LBG	SOM[20]	FS-SOM	MSB-biasedFS-SOM
16	Lena	0.4	96.1	0.4	0.4	0.4
	Baboon	0.4	99.0	0.3	0.5	0.4
	Pepper	0.4	113.4	0.3	0.3	0.4
32	Lena	2.0	232.5	1.2	1.2	1.2
	Baboon	1.2	319.1	1.1	1.3	1.2
	Pepper	1.2	355.4	1.6	1.2	1.2
64	Lena	3.8	563.6	4.0	4.4	4.0
	Baboon	5.7	817.4	4.2	4.7	4.4
	Pepper	4.0	1484.3	4.3	4.3	4.3
128	Lena	13.8	1623.8	18.6	16.5	17.0
	Baboon	19.4	2009.4	18.7	16.7	17.6
	Pepper	16.5	2212.5	16.6	18.9	15.0
256	Lena	56.6	3783.6	61.2	58.8	56.5
	Baboon	57.8	4423.9	56.1	63.0	60.7
	Pepper	57.0	3806.5	77.0	61.0	66.0

achieve a PSNR value of 29.62 dB. Its worst PSNR can drop dramatically down by more than 3 dB to around 26.30 dB. On the contrary, our proposed FS-SOM algorithm has a very flat PSNR response at around 29.60 dB over all combinations of β and k. The difference between the peak and valley of Fig. 13(b) is less than 0.4 dB.

The topology mapping property of FS-SOM and is also studied and compared with the current art SOM in CQ [20] under different sets of parameters. In the 1-D string neighborhood structure, the topological order can also be measured by an objective function J, which is defined as [12]

$$J = \sum_{i=2}^{N} ||w_i - w_{i-1}|| - ||w_N - w_1|| \tag{15}$$

where N is the number of neurons and w_i is the weight vector of the ith neuron. The lower the value of J, the better the codebook is ordered. The results are shown in Fig. 14.

In Fig. 14, we can see that FS-SOM is also robust in its topological ordering performance under different settings of parameters. The J value of SOM is very unstable, varying from 13 to 1173 in Fig. 14(a). Unfortunately, the best topological map of SOM gives the worst quality, as can be seen in Fig. 13(a). On the other hand, the (J) value of our proposed FS-SOM in Fig. 14(b) is very stable, ranging from 293 to 383. FS-SOM gives smaller J than SOM in 24 out of the 30 combinations of β and k, and of the 6 combinations for which SOM has smaller J values, its reconstruction quality happens to be the worst as can be seen from Fig. 13(a).

Because of the topological ordering property of SOM and FS-SOM, the encoded image after the pixel mapping process will still maintain a strong spatial correlation as in the original image. The entropy of the encoded images generated by the 256-neuron SOM and FS-SOM are tested by subjecting their indices to CALIC [26], a competitive lossless image coding algorithm. The encoded images that produce the PSNR values in

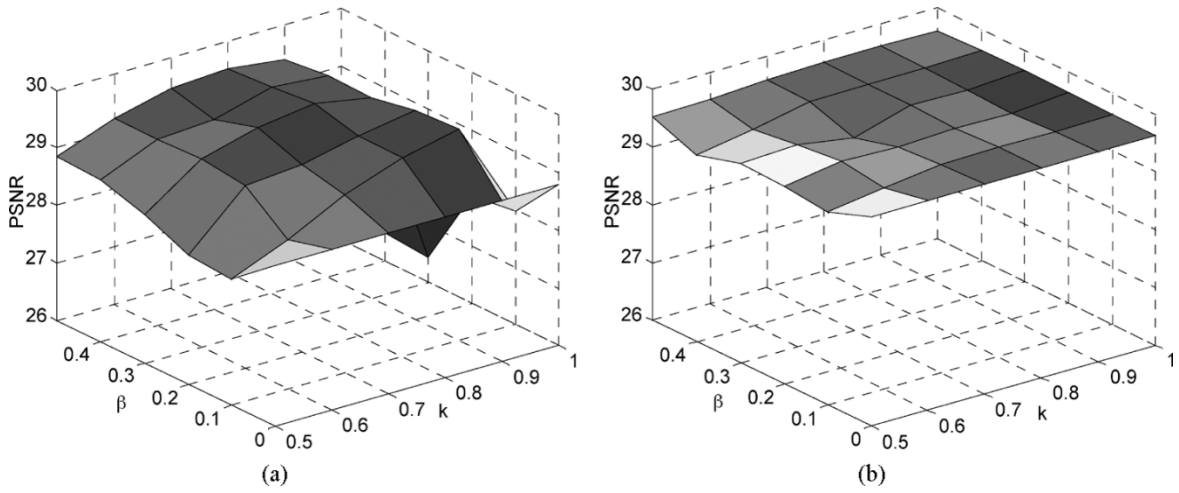


Fig. 13. Palettized image quality under different combinations of β and κ. (a) SOM. (b) FS-SOM.

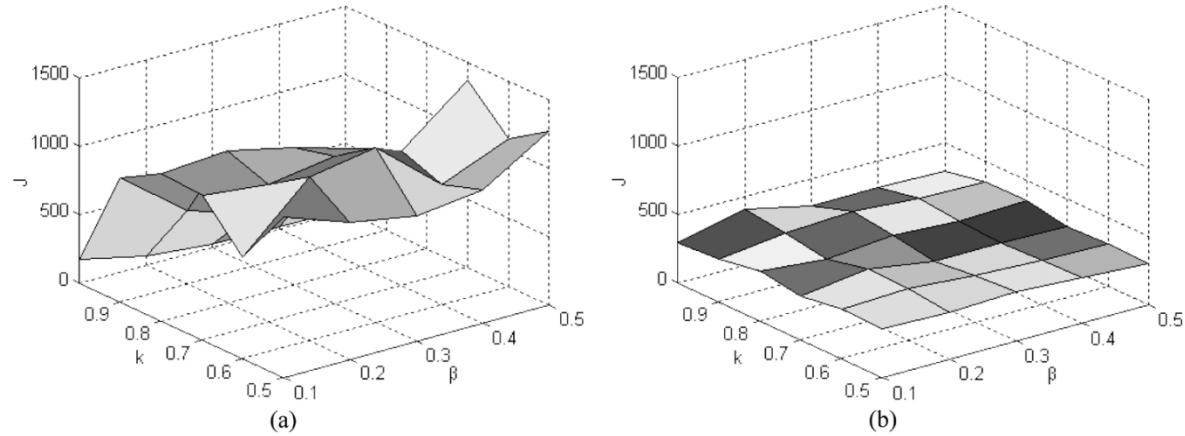


Fig. 14. Topology quality metric, J under different combinations of β and κ. (a) SOM. (b) FS-SOM.

Table I are used. The compression results are shown in Table III. Higher compression ratios are achieved by FS-SOM for all the encoded images. This shows that FS-SOM is able to reduce the bit-rate of the encoded image without scarifying its reconstruction quality, and these merits are sustainable under a wide spectrum of combinations of β and k.

TABLE III
COMPRESSION RATIO OF THE INDEXES OF THE ENCODED IMAGES COMPRESSED BY CALIC

	Lena	Pepper	Baboon
SOM	1.59:1	1.58:1	1.20:1
FS-SOM	1.67:1	1.74:1	1.27:1

V. CONCLUSION

The advents of color photographic processing and multimedia technologies have generated huge archives of digital color images in a vast diversity of applications such as medicine, remote sensing, entertainment, and digital television. Unfortunately, the cost of storage and bandwidth requirements is daunting for many embedded applications that rely on compact detachable memory card and low bit rate data transfer protocols. With the proliferation of wireless mobile terminals, low cost, low capacity color displays are also getting popular nowadays. Using SOM for the CQ problem of these display devices has several added advantages. The SOM network is capable of learning adaptively from input data with an online algorithm, and the algorithm renders itself nicely to a massively parallel architecture for hardware implementation. More importantly, the topology preserving property of SOM yields an ordered color palette, which can be beneficially exploited to speed up the encoding process. In this paper, we have instilled a novel CL scheme into SOM for CQ. The proposed Frequency-Sensitive SOM (FS-SOM) features a harmonious blend of global butterfly-jumping sequence for input data presentation, frequency sensitive learning scheme attuned to neighborhood adaptation and dead neuron reinitialization technique. By localizing the learning rate to each neuron, the output quality of the palettized image has improved significantly. CQ based on the proposed FS-SOM is very efficient and the performance is robust against variation in network parameters compared to the current art SOM. A new MSB-biased encoding scheme has also been introduced to augment the reconstruction performance of FS-SOM. This fast and simple input encoding is used to collapse the RGB space into an octal cube. Each neuron in the octal cube is endowed with a greater capacity to represent a few different color clusters upon decoding. Without increasing the size of the neural network, the MSB-biased encoding scheme effectively circumvents the false contouring artifacts due to the limited

number of available colors. This input encoding is particularly useful in reducing the hardware implementation cost of neural network as a small number of processing elements is needed and the overhead due to the MSB-biased encoding can be lowered by leverage on the existing lossless encoder in a global lossy compression scheme.

REFERENCES

[1] C. Amerijckx, M. Verleysen, P. Thissen, and J. D. Legat, "Image compression by self-organized Kohonen map," *IEEE Trans. Neural Netw.*, vol. 9, no. 3, pp. 503–507, May 1998.

[2] J. P. Braquelaire and L. Brun, "Comparison and optimization of methods of color image quantization," *IEEE Trans. Image Process.*, vol. 6, no. 7, pp. 1048–1052, Jul. 1997.

[3] O. T.-C. Chen, B. J. Sheu, and W.-C. Fang, "Image compression using self-organization networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 5, pp. 480–489, Oct. 1994.

[4] F. L. Chung and T. Lee, "Fuzzy competitive learning," *Neural Netw.*, vol. 7, no. 3, pp. 539–551, 1994.

[5] A. S. Galanopoulos, R. L. Moses, and S. C. Ahalt, "Diffusion approximation of frequency sensitive competitive learning," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 1026–1030, Sep. 1997.

[6] R. M. Gray, *Source Coding Theory*. Norwell, MA: Kluwer, 1990.

[7] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. on Inf. Theory*, vol. 44, no. 6, pp. 2325–2383, Jun. 1998.

[8] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*. Reading, MA: Addison-Wesley, 1991.

[9] R. Hecht-Nielsen, "Counterpropagation networks," *Appl. Opt.*, vol. 26, pp. 4979–4984, 1987.

[10] P. G. Howard, F. Kossentini, B. Martins, S. Forchhammer, and W. J. Rucklidge, "The emerging JBIG2 standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 7, pp. 838–848, Jul. 1998.

[11] T. Kaukoranta, P. Franti, and O. Nevalainen, "A fast exact GLA based on code vector activity detection," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1337–1342, Aug. 2000.

[12] T. Kohonen, *Self-organizing Maps*. Berlin, Germany: Springer-Verlag, 2001.

[13] W. Kou, *Digital Image Compression, Algorithms and Standards*. Norwell, MA: Kluwer, 1995.

[14] A. K. Krishnamurthy, S. C. Ahalt, D. E. Melton, and P. Chen, "Neural networks for VQ of speech and images," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 1, pp. 25–29, Jan. 1992.

[15] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, 1980.

[16] T. M. Martinez, S. G. Berkovich, and K. J. Schulten, "Neural-gas network for vector quantization and its application to time-series prediction," *IEEE Trans. Neural Netw.*, vol. 4, no. 4, pp. 558–568, Jul. 1993.

[17] M. T. Orchard and C. A. Bouman, "Color quantization of image," *IEEE Trans. Signal Process.*, vol. 39, no. 12, pp. 2677–2690, Dec. 1991.

[18] N. Papamarkos, "Color reduction using local features and a SOFM neural network," *Int. J. Imaging Syst. Technol.*, vol. 10, no. 5, pp. 404–409, 1999.

[19] N. Papamarkos, A. E. Atsalakis, and C. P. Strouthopoulos, "Adaptive color reduction," *IEEE Trans. Syst., Man, Cybern. B*, vol. 32, no. 1, pp. 44–56, 2002.

[20] S. C. Pei and Y. S. Lo, "Color image compression and limited display using self-organization Kohonen map," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 2, pp. 191–205, Apr. 1998.

[21] S. C. Pei, C. M. Cheng, and L. F. Ho, "Limited color display for compressed image and video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 6, pp. 913–922, Sep. 2000.

[22] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.l*, vol. 27, pp. 379–423, 1948.

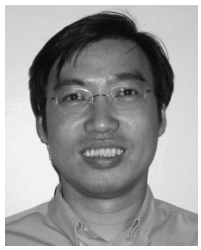
[23] L. G. Shapiro and G. C. Stockman, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 2001.

[24] G. Sharma and H. J. Trussell, "Digital color imaging," *IEEE Trans. Imaging Process.*, vol. 6, no. 7, pp. 901–932, Jul. 1997.

[25] B. Tian, H. X. Tian, and K. C. Yi, "A fast codebook design algorithm for vector quantization," in *Proc. 5th Int. Conf. on Signal Processing (WCCC-ICSP)*, vol. 3, 2000, pp. 2004–2007.

[26] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun.*, vol. 45, no. 4, pp. 437–444, Apr. 1997.

[27] R. Xiao, C. H. Chang, and T. Srikanthan, "On the initialization and training methods for Kohonen self-organizing feature maps in color image quantization," in *Proc. 1st Int. Workshop on Electronic Design, Tes,t and Applications (DELTA-2002)*, Christchurch, New Zealand, 2002, pp. 321–325.



Chip-Hong Chang (S'92–M'98–SM'03) received the B.Eng. (Hons.) degree in electrical engineering from National University of Singapore in 1989, and the M.Eng. and Ph.D. degrees both from the School of Electrical and Electronic Engineering of Nanyang Technological University, Singapore, in 1993 and 1998, respectively.

He was a Component Engineer for General Motors, Singapore, and a Technical Consultant of Flextech Electronics Pte. Ltd. He was as a Lecturer in the Electronics Design Center, Nanyang Polytechnic University, Singapore, in 1993. Since 1999, he has been with the School of Electrical and Electronic Engineering, Nanyang Technological University, where he is currently an Assistant Professor. He holds concurrent appointments as the Deputy Director of the University level Research Center, Center for High Performance Embedded Systems, and the Program Director of VLSI Design and Embedded Systems research group of the Center for Integrated Circuits and Systems. His current research interests include low power arithmetic circuits, design automation and synthesis of digital filters, self-organizing maps and algorithms and architectures for digital image processing. He has published more than 90 refereed international journal and conference papers, and book chapters.

Pengfei Xu received the B.Eng. degree from Xi'an Jiaotong University, Xi'an, China in 2002. He is currently pursuing the Ph.D. degree in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

His research interests include data clustering and image segmentation.

Rui Xiao received the B.Eng. and M.Eng degrees (both with first class honors) from the Nanyang Technological University, Singapore, in 2002 and 2004, respectively.

She is currently working as a Digital IC Designer at STMicroelectronics, Singapore. Her research interest is in digital image processing.

Thambipillai Srikanthan received the B.Sc. degree (Hons.) in computer and control systems and the Ph.D. degree in system modeling and information systems engineering from Coventry University, Coventry U.K.

He has been with the Nanyang Technological University (NTU), Singapore, since 1991, where he holds a joint appointment as Associate Professor and Director of the Center for High Performance Embedded Systems. His research interests include System integration methodologies, architectural translations of compute intensive algorithms, high-speed techniques for image processing and dynamic routing. He has published more than 100 technical papers and has served a number of administrative roles during his academic career. He is the founder and Director of the Center for High Performance Embedded Systems, which is now a university-level research center at NTU.

Dr. Srikanthan is a corporate member of the Institution for Electrical Engineers, U.K.