Adaptive Extraction of Principal Colors Using an Improved Self-Growing Network

Yurong Li
School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu 610074, China
Chengdu Institute of Computer Application, the Chinese Academy of Science, Chengdu 610041, China
liyr_t@swufe.edu.cn

Zhengdong Du
Department of Mathematics, Sichuan University, Chengdu, Sichuan 610064, China
zdu85@yahoo.com, fu_hongguang@hotmail.com

Hongguang Fu
Chengdu Institute of Computer Application, the Chinese Academy of Science, Chengdu 610041, China
zdu85@yahoo.com, fu_hongguang@hotmail.com

***Abstract*—This paper aims to solve the two major issues existing in current color quantization algorithms. The first one is to require users to specify the number of representative colors in advance; the other is that it is difficult in choosing the colors to describe accurately the essential details represented by small groups of pixels isolated in the color space. Based on the growing mechanism of the Growing When Required neural network, a novel algorithm is proposed to adaptively extract the prominent colors of an image. A number of criteria are introduced that have an effect on controlling of the number and topology of neurons in the output layer. A global permutation method to rearrange the input sample order is presented based on Linear Pixels Shuffling in order to improve the performance of the network. The experiments show that the proposed method can automatically estimate the number of colors to efficiently represent an original image, meanwhile capable of retaining important isolated colors even when the number of the representative colors is low. It is also shown that the algorithm outperforms the popular ones in terms of color distortion.**

***Index Terms*—color quantization, incremental learning, self-growing network, neural network, Linear Pixel Shuffling**

I. INTRODUCTION

Color quantization, also known as color reduction, is a process of reducing the number of colors in an image to a low number of representative colors while keeping the perceived difference between the original image and its quantized image as low as possible. It is an important technology in digital image processing. A digital image is usually described by a set of pixels distributed in two-dimensional grid. A true-type color image consists of more than 16 million different colors in a 24-bit RGB color space. It is easier to understand and process an image with a low number of colors. It is preferable that images are quantized to as low number as possible. So it is needed for color quantization in transmission, compression, presentation, segmentation or retrieval of visual information, and so on.

Many techniques have been developed for color quantization. They are classified in major two categories. The first class of algorithms is based on splitting algorithms in which the color space is divided into disjoint regions by consecutively splitting up the color space. For every region, a color is chosen to represent it. Median-Cut [1], the variance-based method [2] and the Octree algorithm [3] belong to this class. The idea behind Median-cut is to have every chosen color represent the same number of pixels in an original image. It splits the color space into two cubes along the longest axes in such a way that nearly equal numbers of pixels is in each part. The cube to be divided is chosen as the one with the most pixels. The process is repeated until the desired number of cubes is generated. The variance-based algorithm is similar to Median-Cut, the difference is the criterion of splitting. The cube is split at the position where the variance of the marginal distribution is minimized. In the Octree algorithm, a color space is cut into smaller regions based on a data structure, named Octree, and the neighboring regions are repeatedly merged. A major disadvantage of these algorithms is to ignore the interrelationship between neighboring color regions in the process of split.

To resolve the problem mentioned above, the second class of algorithms is proposed which depends on clustering analysis. Artificial neural network is a very powerful tool to resolve clustering. The Kohonen Self-Organized Feature Map (KSOFM) neural network [4] is the most widely used. KSOFM consists of two layers of fully connected neurons, the input layer and competitive layer. Both the number of the neurons and connections are predefined and unchanged in the process of network training. Generally it can capture the topology and distribution of input samples for data analysis. Several

doi:10.4304/jcp.5.2.202-209

implementations of KSOFM have been proposed for uses one-dimensional KSOFM every neuron of which is for a cluster. Through the learning, the weight vectors are obtained which correspond to the representative colors. An extension of the KSOFM-based method is the algorithm [6] which sub-samples pixels in an image in a smooth way and updates only the closest neighboring clusters in order to significantly speed up the algorithm with slightly degraded quantization results. Ref. [7] employs a developed network which combines the KSOFM network and a supervised counter propagation neural network. The latter network is applied to find the average value of a color cluster.

In general, the algorithms based on clustering analysis have better quantization results than ones based on splitting scheme. But it has much higher complexity and cost much more running time than the latter ones.

Among the existing algorithms of the two categories, two major issues are needed to be resolved. One is that users have to preset the number of representative colors. It is difficult for users to estimate the number of the principal colors to efficiently describe an arbitrary image. Numbers of the colors which can efficiently represent different images are various because of variable complexities of images, as numbers of colors in different original images are diverse. The other is that small groups of pixels isolated in the color space suffer being neglected when the number of representative colors is reduced. For example, as it can be observed in Fig. 1, the number of the pixels is very small which correspond to the red cloth. The red color is obviously different from the around colors, which is important for the idea of this image. So it is called the important isolated colors. The red color is evidently changed when the original image is quantized to 16 colors by a traditional quantization method, as shown in Fig. 2. The change of the color causes that the quantized image is obviously different from its original one. In additional, it is ineffective for image analysis and segmentation to lose the important isolated color. The algorithms mentioned above consider only the distance and distribution of colors in a color space, resulting in ignoring the uncommonly colors.

A few algorithms have been proposed to solve the two issues. The DWT-based method [8] determines the number of representative colors by using the discrete wavelet transform, but it leave the second issue out. An color quantization. The KSOFM-based algorithm [5] adjustable algorithm [9] retains the important isolated colors by involving user adjusted weights, but it has to preset the number of representatives. The method in Ref. [10] has tried to solve the two problems. It combines the features of the KSOFM and Growing Neural Gas (GNG) networks, and input the union of the color information and spatial features to the developed neural network. It can estimate the number of the colors. But in practice, as we can see it in Fig. 2, the second problem still exists when the number of the representative colors resulted by this algorithm is very low.

In this work, an algorithm based on the growing mechanism of the Growing When Required network (GWR) [11] is proposed which adaptively adjusts the number of neurons, i.e., the number of the representative colors, according to the structure of input data. Three criteria are applied to control the growing of the neurons. Beside the number, connections of the neurons in the network are adjustable. A method of arrangement for the input sample is presented based on Linear Pixels Shuffling (LPS) [12] in order to improve the performance of the neural network.

The rest of this paper is organized as follows. Section 2 introduces the method of sampling pixels of an image in a smooth way. Section 3 mainly introduces the developed color quantization algorithm based on the GWR network. The preliminary results are given in Section 4. The conclusion is drawn in Section 5.

II. IMAGE SAMPLING

The input samples are obtained by examining all pixels in an image, diving the scan into a number of cycles. A data set is defined every scanning cycle. In order to improve the performance of the neural network, the learning procedure requires the input vectors to be randomly distributed [4] and the data sets to be similar. This is done by using the developed sampling pixels method based on Linear Pixels Shuffling which visits pixels of an image in a smooth manner [6]. The visiting is designed so that every sub-rectangle of an image contains roughly equal number of the pixels which are evenly distributed over the whole image.

Let us consider a numerical sequence called G defined as follows:

Figure 1. Original alley.jpg

Figure 2. Quantized image

$$G_0 = 0, \quad G_1 = 1, \quad G_2 = 1$$
$$G_n = G_{n-1} + G_{n-3}, \ n \geq 3 \tag{1}$$

The first few items are: 0, 1, 1, 1, 2, 3, 4, 6, 9, 13, 19, 28, 41, 60, 88, 129, 189, 277, 406, 595, ... Suppose that parameters *A*, *B* and *C* are three successive numbers in the sequence *G*.

Let us consider an image of size $w \times h$. The image is rounded to a square of size $G_n \times G_n$, $G_n \geq w$ and $G_n \geq h$. The pixels outside of the image are ignored. A $G_n \times G_n$ mask table *M* with entries is defined as follows:

$$M_{xy} = (xA + yB) \ \% \ C \tag{2}$$

Where x and y denote the position of a pixel.

For all elements in the mask table, every number in 0, 1, 2, 3, 4, …, G_n-1 occurs exactly G_n times. Fig. 3 is a portion of the mask table using A=60, B=88 and C=129. As it can be observed in Fig. 3, the values in the table which are numerically close are geometrically distant.

(0)	88	47	6	94	53	12	100	59	18	106	65	24
60	19	107	66	25	113	72	31	119	78	37	125	84
120	79	38	126	85	44	3	91	50	9	97	56	15
51	10	98	57	16	104	63	22	110	69	28	116	75
111	70	29	117	76	35	123	82	41	(0)	88	47	6
42	1	89	48	7	95	54	13	101	60	19	107	66
102	61	20	108	67	26	114	73	32	120	79	38	126
33	121	80	39	127	86	45	4	92	51	10	98	57
93	52	11	99	58	17	105	64	23	111	70	29	117
24	112	71	30	118	77	36	124	83	42	1	89	48
84	43	2	90	49	8	96	55	14	102	61	20	108
15	103	62	21	109	68	27	115	74	33	121	80	39
75	34	122	81	40	128	87	46	5	93	52	11	99

Figure 3. Portion of the mask table

In our algorithm, the pixels are firstly taken which correspond to the positions (*x*, *y*) for which $M_{xy} = 0$, followed by the positions (*x*, *y*) for which $M_{xy} = 1$, and so on. G_n pixels are examined each scanning cycle. The collection of G_n samples defines a training data set, which correspond to the positions for which the values of M_{xy} are same. Thus G_n different data sets are constructed. The corresponding samples are taken from neighboring pixels, so the data sets are considered to be similar. All the training data sets are circularly input to the network until the neural network convergences.

III. COLOR QUANTIZATION

A. The improved GWR network

The Growing When Required network [11] is a self-growing network. In contrast with the KSOFM network, the size and topology of the GWR network are not fixed or predefined. It starts with two neurons, and then the number is progressively increased. A new neuron can be added at any time whenever an input vector is not sufficiently matched to the winner. The network growing stops once it has matched the input data. The connections between two neurons define the neighborhoods of nodes. The connections are also created or deleted dynamically during training process. The GWR network enables preservation of the topology of input data.

In this work, the GWR network is used to define the mapping from color components of pixels of an original image to the closet representative colors. The structure of the network is not needed to predefine, so the number of the representative colors is not necessary to be determined in advance. The final number of the output neurons defines the number of the principal colors for the original image.

The network consists of two layers of fully connected: input layer and output layer. The samples taken from an original image are input to the network. The weight vectors of the output neurons correspond to the representative colors. The neurons of the output layer are dynamically created or deleted. Three criteria are used to control the growing of the lattice of output layer. The first criterion is to insert a new neuron near this one which is the best matching neuron to an input vector when the winner is not sufficiently matched to the input vector and it is a well-trained neuron, according to predefined conditions. The second one is to remove the neurons that have no neighboring neurons. Finally, remove the neurons that enough close to their neighbors. In the process of the network training, the weight vectors are updated in such a way that the vectors move to the input vectors. Unlike the KSOFM network, the learning rate of the weight vectors is based on the fired frequency of the neurons. The connections between the output neurons are also created or eliminated dynamically. Only the neurons directly linked to the updated winner neurons will be updated. The network training will stop when the change of the number of the output neurons is stable, that is, the difference of the numbers is within a predefined value.

B. Training of the self-growing network

For every neuron $i \in [1, n]$ of the lattice of neurons, a weight vector w_i, accumulated error variable ERR_i, a global counter $N_i^{(1)}$ and a local counter $N_i^{(2)}$ are defined. w_i expresses the color components of a representative color. ERR_i represents quantity of total color quantization error that corresponds to neuron *i*. $N_i^{(1)}$ and $N_i^{(2)}$ denote the number of the times when neuron *i* becomes the best matching neuron in the whole training period and the current epoch, respectively. For a lateral connection between two adjacent neurons *k* and *h*, variable $age_{(k, h)}$ is defined which describes the age of this connection. If $age_{(k, h)} \geq 0$, there is a connection between the neurons. If $age_{(k, h)} = -1$, the two neurons are disconnected. In the following equations, ||.|| denotes the Euclidean distance.

Initially, the output layer consists of two neurons. The values for weight vectors w_i (i = 1, 2) are initialized by randomly selecting two different input vectors. Let $N_i^{(1)} = 0$, $N_i^{(2)} = 0$, $ERR_i = 0$, $i \in [1, 2]$. All the training data sets are repeatedly input to the network. Each time a data set is fed, the number of the training epochs increases by one. The training steps for the network consisting of *n* output neurons are as follows:

Step 1: At the beginning of every epoch, accumulated error ERR_i and local counter $N_i^{(2)}$ $i \in [1, n]$ are reset to zero.

Step 2: For a given input vector p, the first winner neuron b and second winner neuron s are obtained according to the following relations:

$$b = \arg\min_{i\in[1,n]} \| p - w_i \| \quad (3)$$

$$s = \arg\min_{i\in[1,n]/\{b\}} \| p - w_i \| \quad (4)$$

Step 3: The accumulated error ERR_s of the second winner neuron is updated:

$$ERR_s = ERR_s + \| p - w_s \| \quad (5)$$

Step 4: The lateral connection is created if there is not edge between neuron b and neuron s. Let $age_{(b,\ s)} = 0$.

Step 5: The similarity between p and the winner neuron b is calculated according to the following equation:

$$sim = e^{-\|p-w_b\|/\sqrt{3*255*255}} \quad (6)$$

Step 6: Let us consider a thresh $T_{sim} \in (0, 1)$. If $sim \geq T_{sim}$, it is considered that the winner neuron b is well matched with p, otherwise, it is necessary to think whether the winner neuron is well trained, since the winner has not been trained well to match the coming input correctly. The algorithm employs a simple method to tell whether a neuron is trained well by using a counter recording how often a node is fired. If the value of the global counter $N_i^{(1)}$ for a neuron i is greater than a predefined value T_{num}, it is considered to be well trained. The input vector is considered not sufficiently matched with the winner neuron, if the following conditions are hold: $sim < T_{sim}$ and $N_b^{(1)} \geq T_{num}$.

Step 7: If the winner is not well matched with the input vector p, a new neuron r is inserted into the lattice of the neurons of the output layer, which is close to the winner neuron b. The weight and the counters of the new neuron are set according to following equations:

$$w_r = (w_b + p)/2 \quad (7)$$

$$N_r^{(1)} = 1, N_r^{(2)} = 1 \quad (8)$$

The lateral connection between the neuron b and s is deleted. Let $age_{(b,\ s)} = -1$. The connections between the new neuron and neuron b and s are created, respectively. Let $age_{(r,\ b)} = 0$ and $age_{(r,\ s)} = 0$.

Step 8: If no new neuron is added, the winner neuron b is adapted. In accordance with the GWR network [5], the weight is modified according to the following relations:

$$\Delta w_b = \varepsilon_b \times \eta_b \times (p - w_b) \quad (9)$$

$$\eta_b = 1 - \frac{1}{\alpha_b}(1 - e^{(-\alpha_b * N^{(1)}/\beta_b)}) \quad (10)$$

Where $0 < \varepsilon_b < 1$, $\alpha_b = 1.05$ and $\beta_b = 3.33$. The learning rate η_b is reduced according to the global counter $N_i^{(1)}$.

Step 9: The weight vectors of the neurons connected to the adapted winner neuron are correspondingly adjusted, according to the following equations:

$$\Delta w_a = \varepsilon_a \times \eta_a \times (p - w_a) \quad (11)$$

$$\eta_a = (1 - \frac{1}{\alpha_a}(1 - e^{(-\alpha_a * N_a^{(1)}/\beta_a)})) * e^{(-age^2/T_{age})} \quad (12)$$

Where $0 < \varepsilon_a < 1$, $\alpha_a = 1.05$, $\beta_a = 14.3$ and $T_{age} = 500$. The learning rate η_a is reduced, according to the global counter, as well as the quantity $age_{(a,\ b)}$.

The ages of the connections between the winner and the neighboring neurons i, excepting the second winner are increased by one. Let $age_{(i,\ b)} = age_{(i,\ b)} + 1$.

Step 10: At the ending of each epoch, with the purpose of removing superfluous neurons and connections, some neurons and connections are deleted according to following criteria: The first criterion is to remove the neurons without neighboring neurons; second one removes the connection the age of which is greater than 500; the last one removes the neurons enough close to their neighbors. The neuron with the minimum $ERR_i/N_i^{(2)}$ which expresses the average distance to its neighbors is deleted if the following condition is satisfied:

$$ERR_i / N_i^{(2)} \leq \gamma \times \frac{2}{n(n-1)} \sum_{\forall(i,j)\in[1,n]} \| w_i - w_j \| \quad (13)$$

Where $\gamma \in [0, 1]$ is a user-defined thresh.

Ⅳ. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed algorithm, six standard colored images [13] and the image alley.jpg shown in Fig. 1 are used, whose basic information are shown in Table Ⅰ.

The RGB color space is used in our experiment in order to have comparable results. In all the following cases, let $\varepsilon_b = 0.1$, $\varepsilon_a = 0.005$, in (9) and (10), respectively. The threshold T_{num} is set to 10. The threshold for removing neurons takes $\gamma = 0.1$.

A. Experiment 1

The first experiment evaluates the ability of the developed algorithm to estimate of the number of the representative colors for an arbitrary image. Three color images are quantized to demonstrate the result. These are

TABLE I. INFORMATION OF THE TESTED IMGAGE

Images	Size	Color Number
palm.png	488×724	69095
im10.png	256×256	29091
girl.png	512×512	22692
lenna.png	512×512	148279
mandrill.png	507×509	256
peppers.png	512×512	111344
alley.jpg	447×299	61390

representative colors for an arbitrary image. Three color images are quantized to demonstrate the result. These are "palm.png", "girl.png" and "im10.png" images, which are depicted in the left column of Fig. 4.

To represent original images with perceived difference between original and their quantization images as low as possible, the threshold Tsim takes a greater value, Tsim = 0.85.

The training procedure will stop when the difference of the numbers is within a predefined value *Vnum* among consecutive *K* training epochs. Let *Vnum* = 5 and *K* = 5.

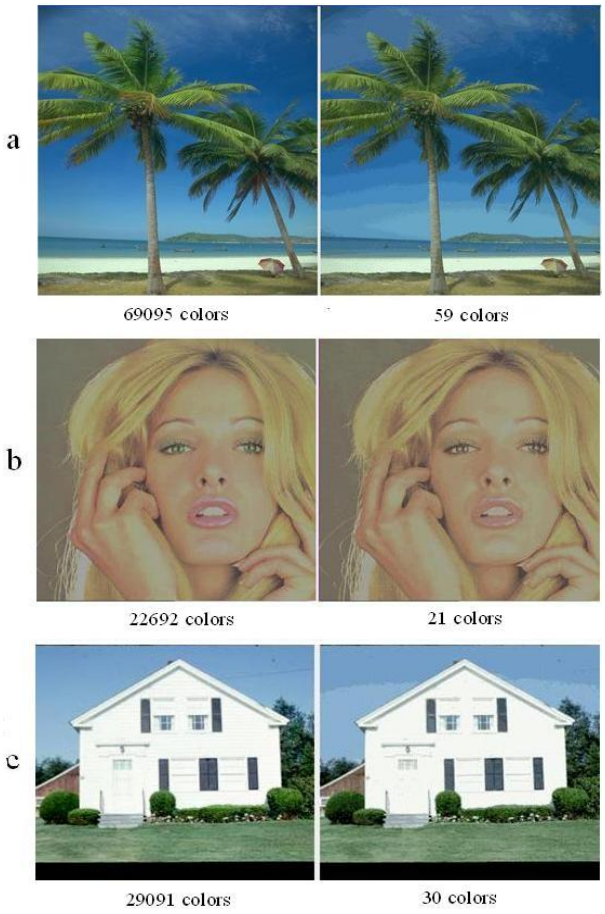

Figure 4. Original images are depicted in left column. Adaptive quantized results are depicted in the right one. (a) palm; (b) girl; (c) im10

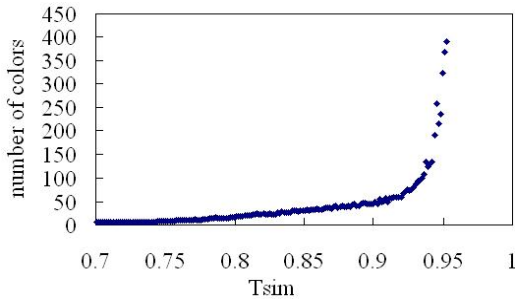

Figure 5. Changing of colors number following T_{sim}

The quantized results are depicted in the right column of Fig. 4. The numbers of the representative colors of the test images decided by the proposed algorithm are 59, 21 and 30, respectively. In the terms of the human perception, the quantized images are very close to the corresponding original ones. The results show that our algorithm can estimate the number of the principal colors according to an arbitrary image.

As the result is shown, the number of the principal colors in a quantized image mainly depends on its original image. Besides, the value of parameter Tsim has important effect on it. Taking im10.png for example, Tsim is set to different values when the image is quantized. The result is given in Fig.5. More the value of Tsim is, more the number of colors is. The value of Tsim decides how similar the quantized images are to corresponding original images.

B. Experiment 2

Experiment 2 tests the ability to retain the important isolated colors. The image alley.jpg in Fig. 1 is quantized to demonstrate the result. Except the developed algorithm, other color quantization methods, including Median-Cut, the Octree and KSOFM-based method [5],

Fig. 6, the "red" pixels become "green" ones in the quantized images by the Median-Cut and KSOFM-based algorithm. The Octree method has better result. The proposed method has achieved the best result, which is the closet to the original image in visual perception.

C. Experiment 3

The proposed algorithm is compared with several well-known color quantization methods: Median-Cut [1], the Octree [3] and KSOFM-based method [5] in terms of statistical measures MSE, SNR, PSNR.

Let us consider a data set *PX* consisting of *N* input vectors. If $p_i \in PX$ is an input vector, and p'_i the corresponding quantized vector, (14), (15) and (16) are used to measures of the statistical color quantization errors.

$$MSE = \frac{1}{N}\sum_{i=0}^{N}(p_i' - p_i)^2 \tag{14}$$

$$SNR = 10\log\left(\frac{\sum_{i=0}^{N} p'^{\,2}_i}{\sum_{i=0}^{N}(p'_i - p_i)^2}\right) \tag{15}$$

$$PSNR = 10\log\left(\frac{3\times 255\times 255\times N}{\sum_{i=0}^{N}(p_i' - p_i)^2}\right) \tag{16}$$

The set of the images are used whose information is described in Table Ⅰ. For the comparison reason, these images are respectively quantized to 16, 32 and 64 unique colors. The value of Tsim is adjusted so that the number of the neurons convergences to 16, 32 and 64, respectively. To avoid the convergence criterion influencing quantization result, the training procedure stops exactly after all pixels of an image are fed to the

network 15 times in all cases in the developed algorithm.

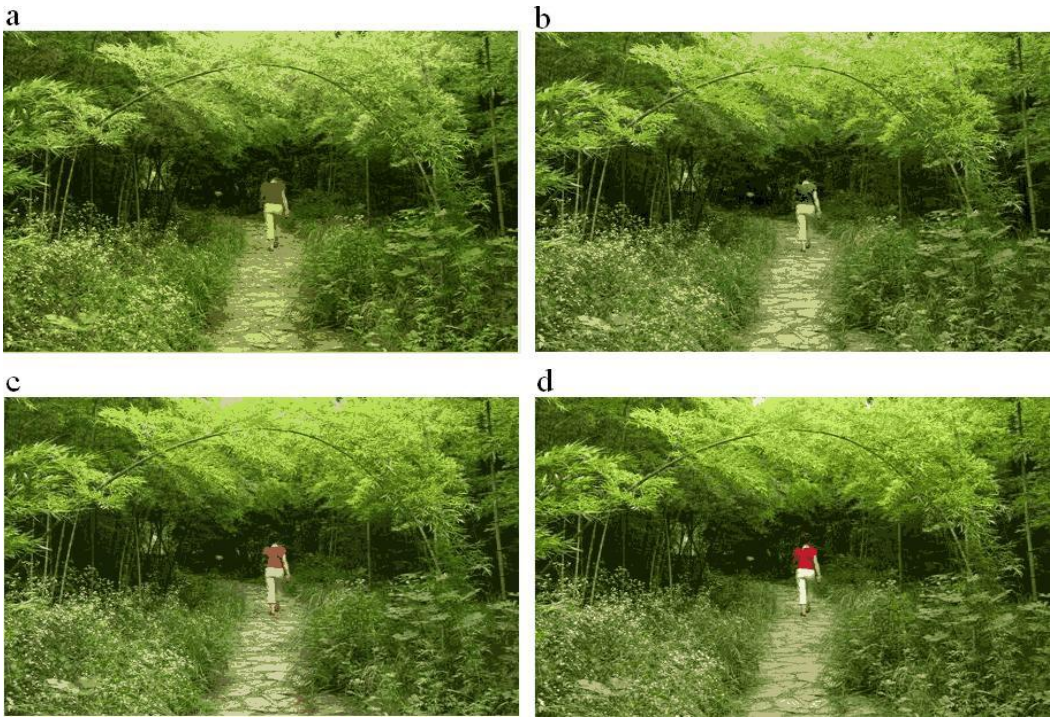

Figure 6. Quantized images by different techniques: (a) Median-Cut; (b) KSOFM-based method; (c) Octree method; (d) the proposed algorithm

The comparative results of Table Ⅱ– Ⅳ correspond to the quantization of the original images to 16, 32 and 64 colors, respectively. In every table, the developed algorithm is compared with other quantization methods in the terms of the quality measures MSR, SNR and PSNR. As it can be seen in Table Ⅱ– Ⅳ, the proposed algorithm achieves significant reduction in MSE and improvement in SNR and PSNR, compared with the

TABLE II Comparison of quantized results, original images are quantized to 16 colors

		palm	girl	im10	lenna	mandrill	peppers	alley
Ours	MSE	284.8	75.7	182.8	198.7	643.5	382.7	274.3
	SNR	49.3	65.2	61.4	55.3	45.4	48.0	47.1
	PSNR	65.3	78.5	69.7	67.9	57.1	62.3	65.7
KSOFM	MSE	393.8	154.1	257.9	281.3	738.1	515.4	473.0
	SNR	46.1	58.2	58.1	53.6	44.1	45.0	41.5
	PSNR	62.1	71.4	66.3	65.4	55.8	59.4	60.2
Octree	MSE	387.0	262.0	213.7	278.5	759.0	504.5	387.6
	SNR	46.2	52.8	59.9	53.6	43.8	45.2	43.6
	PSNR	62.2	66.1	68.2	65.5	55.5	59.6	62.2
MC	MSE	863.6	138.0	296.8	302.6	942.1	615.2	612.8
	SNR	37.2	59.1	56.6	52.7	41.4	42.9	38.6
	PSNR	54.2	72.5	64.9	64.7	53.3	57.6	57.6

TABLE III Comparison of quantized results, original images are quantized to 32 colors

		palm	girl	im10	lenna	mandrill	peppers	alley
Ours	MSE	158.9	43.5	98.9	122.9	325.3	227.1	166.9
	SNR	55.2	70.7	67.6	61.8	52.2	53.2	52.2
	PSNR	71.1	84.1	75.9	73.7	64.0	67.5	70.6
KSOFM	MSE	195.4	52.1	126.0	145.9	369.4	266.0	201.0
	SNR	53.2	67.6	65.2	60.1	51.0	51.7	50.3
	PSNR	69.1	81.0	73.4	72.0	62.7	66.0	68.8
Octree	MSE	213.8	241.8	108.2	166.1	444.9	292.4	254.7
	SNR	52.4	53.6	66.7	58.8	49.2	50.7	47.9
	PSNR	68.2	66.9	75.0	70.7	60.8	65.0	66.4
MC	MSE	246.9	95.5	179.5	181.8	515.1	357.6	349.7
	SNR	50.7	62.9	61.6	57.8	47.6	48.5	44.5
	PSNR	66.7	76.2	69.9	69.8	59.4	63.0	63.2

TABLE IV Comparison of quantized results, original images are quantized to 64 colors

		palm	girl	im10	lenna	mandrill	peppers	alley
Ours	MSE	95.9	23.5	65.4	88.2	164.7	141.6	146.6
	SNR	60.3	76.9	71.7	65.1	59.1	58.0	53.5
	PSNR	76.2	90.2	80.0	77.1	70.8	72.3	72.0
KSOFM	MSE	109.6	35.7	59.6	83.9	175.8	159.1	109.9
	SNR	59.0	74.4	72.7	65.7	58.5	56.9	56.4
	PSNR	74.8	87.9	80.9	77.5	70.11	71.1	74.8
Octree	MSE	121.1	192.3	53.3	97.5	205.9	169.5	120.5
	SNR	58.0	55.8	73.8	64.2	56.9	56.3	55.5
	PSNR	73.8	69.2	82.0	76.0	68.5	70.5	73.9
MC	MSE	158.4	51.6	102.1	113.7	303.8	211.2	226.6
	SNR	55.2	69.0	67.3	62.5	52.9	53.9	49.0
	PSNR	66.7	82.4	75.5	74.5	64.6	68.3	67.6

Median-Cut, the Octree and KSOFM-based method.

V. CONCLUSIONS

In this paper, a new color quantization algorithm is proposed which utilizes the growing mechanism of the GWR neural network. A global permutation method to rearrange the input data order is presented in order to improve the performance of the network training. The worthy point of our algorithm is that it is able to automatically estimate the number of the principal colors, that is, adaptive quantize an image. In addition, the algorithm is capable of retaining important isolated colors even when images are quantized to a low number of representative colors. The experiments indicate that the proposed algorithm significantly outperforms the widely used Median-Cut, the Octree and KSOFM-based method in terms of measures of the quality measures MSE, SNR and PSNR.

ACKNOWLEDGMENT

This work was supported in part by the Scientific Research Fund of Southwestern University of Finance and Economics, under grant No.YB0806, and by a grant from the "Project 211 (Phase Ⅲ)" of the Southwestern University of Finance and Economics.

REFERENCES

[1] P. Heckbert, Color image quantization for frame buffer display. Computer & Graphics, 1982, 16: 297–307.
[2] S.J. Wan, P. Prusinkiewicz, S.K.M. Wong, Variance based color image quantization for frame buffer display. Color Research and Application, 1990, 15 (1), 52 – 58.
[3] M. Gervautz, W. Purgathofer, A simple method for color quantization: Octree quantization. Proc. CGI 88, 219–231, 1988.
[4] T. Kohonen, Self-Organizing Maps, second ed. Springer, Berlin, 1997.
[5] A. Dekker. Kohonen neural networks for optimal color quantization. Network: Computation in Neural Systems, 1994, 5: 351-367.
[6] Y. Li, Color quantization algorithm based on neural network and linear pixel shuffling, *Opto-Electronic Engineering*, 2007, 34 (9): 124-128
[7] A. Rahman, C.M. Rahman, A new approach for compressing color images using neural network. In: Proceedings of CIMCA 2003, Vienna, Austria, pp. 12 – 14.

[8] N. Kim, N. Kehtarnavaz, DWT-based scene-adaptive color quantization, Real-Time Imaging , 2005, 11: 443–453
[9] B. Zhou, J. Sun, Q. Peng, An adjustable algorithm for color quantization. Pattern Recognition Letters, 2004, 25: 1787-1797
[10] A. Atsalakis, N. Papamarkos, Engineering Applications of Artificial Intelligence, 2006, 19: 769–786
[11] S. Marsland, J. Shapiro, U. Nehmzow, A self-organizing network that grows when required. Neural Networks, 2002, 15: 1041 − 1058.
[12] P. Anderson, J. Arney, K. Ayer, Linear Pixel Shuffling (I): New Paradigms for New Printers. Proceedings of the 16th International Conference on Digital Printing Technologies (NIP16). Vancouver: The Society for Imaging Science and Technology, 2000, 801-806.
[13] Ohio State University. Signal Analysis and Machine Perception Laboratory, Department of Electrical Engineering. http://sampl.ece.ohio-state.edu/database.htm

Yurong Li was received M.S. degree in Photoelectronics Technology Science from Sichuan University, Chengdu, China, in 1995, M.S. degree in computer science from Rochester Institute of Technology, Rochester, USA, in 2002. She served as a senior software engineer in Institute of Technology Physics of Southwestern China and Aurora Consulting Group, Inc. Since 2004, she has joined to the faculty of information engineering department, Southwestern University of Finance and Economics located in Chengdu, Sichuan Province, China. She is currently an associated professor in the department. Her current research interests include digital image processing, dynamical systems and information retrieval. She has published more than 10 journal papers and one textbook.

Zhengdong Du was obtained his B. S. degree in mathematics from Huazhong University of Science and Technology, Wuhan, China in 1989, M. S. degree in computer science from State University of New York at Buffalo, USA in 1998 and Ph. D. degree in mathematics from State University of New York at Buffalo, USA in 2000. He was a software engineer in Analysis and Simulation Inc. located in Buffalo, New York, USA from September 1998 to December 1998 and a software engineer in Aurora Consulting Group Inc. located in East Aurora, New York, USA from January 1999 to December 2002. He came to Department of Mathematics, Sichuan University located in Chengdu, Sichuan Province, China in March 2003 and currently he is an associate professor of SCU. His main interests of research include dynamical systems, symbolic computations and image processing. He has published over 10 research papers and one textbook..

Hongguang Fu was received the B.S. and Ph.D. degree in computational mathematics from Sichuan University located in Chengdu, Sichuan Province, China, in 1986 and 1999, respectively. He holds a joint appointment as Research Fellow and Deputy Director of Chengdu Institute of Computer Application, the Chinese Academy of Science located in Chengdu, China. He is a concurrent professor of the department of computer science in University of Electronic Science and Technology of China. His research areas include symbolic computation and digital image processing. He has published over 30 papers.

Dr. Fu is a member of both IEEE and ACM.