Adaptive Color Reduction

Nikos Papamarkos, *Member, IEEE*, Antonis E. Atsalakis, and Charalampos P. Strouthopoulos

***Abstract*—This paper proposes a new algorithm for the reduction of the number of colors in an image. The proposed adaptive color reduction (ACR) technique achieves color reduction using a tree clustering procedure. In each node of the tree, a self-organized neural network classifier (NNC) is used which is fed by image color values and additional local spatial features. The NNC consists of a principal component analyzer (PCA) and a Kohonen self-organized feature map (SOFM) neural network (NN). The output neurons of the NNC define the color classes for each node. The final image not only has the dominant image colors, but its texture also approaches the image local characteristics used. Using the adaptive procedure and different local features for each level of the tree, the initial color classes can be split even more. For better classification, split and merging conditions are used in order to define if color classes must be split or merged. To speed up the entire algorithm and reduce memory requirements, a fractal scanning subsampling technique is used. The method is independent of the color scheme, it is applicable to any type of color images, and it can be easily modified to accommodate any type of spatial features and any type of tree structure. Several experimental and comparative results, exhibiting the performance of the proposed technique, are presented.**

***Index Terms*—Color quantization, color segmentation, neural networks (NN), principal component analyzer (PCA), self-organized ferature map (SOFM).**

I. Introduction

NOWADAYS, color images occupy an extensive area of the information used in computer technology. True type color images consist of more than 16 million (2^{24}) different colors in a 24 bit full RGB color space. There are many reasons for the reduction of the number of colors in a digital image. Reduction of the number of the image colors is an important task for presentation, transmission, segmentation, and compression of color images. In most cases, it is easier to process and understand an image with a limited number of colors.

The most usually used techniques for color reduction in a digital image are the color quantization and the multithresholding approaches. The color quantization techniques group similar colors and replace them with only a single "quantized" color [1], [5], [6], [8]. The ultimate goal of the classical color quantization techniques is to reduce the number of colors of an image with minimum distortion [28]. Therefore, the main objective of computer graphics research in the color quantization area is to select an optimal palette that ensures minimization of a perceived difference between the original and the quantized images [29].

Several techniques have been proposed for color quantization. First, there is a class of techniques that are based on splitting algorithms. According to these approaches, the color space is divided into disjoint regions, by consecutively splitting up the color space [23]. In this category belong the methods of octree [1], [6], median-cut (MC) [8], and variance-based algorithm [30]. In another major class of color quantization algorithms belong methods based on cluster analysis. The frequently clustering techniques used in this category are the Kohonen SOFM [5], Fuzzy C-means [14], [24], C-means [24], and K-means [29]. The above techniques are suitable for eliminating the uncommon colors in an image but they are ineffective for image analysis and segmentation. The quality of the resultant image varies depending on the number of the final colors and the algorithm intelligence. Generally, a color quantization algorithm consists of two main stages: the color palette generation stage and the pixel mapping stage [9]. Color quantization techniques are usually used for the reduction of the colors of an image to a large number of colors. For example, from 16 million to 256 colors. These techniques are not good enough to produce images with a very limited number of colors. As it is noticed by Buhmann *et al.* [22], one of the basic disadvantages for all color quantization approaches is the fact that they neglect spatial, i.e., contextual, information. That is, until now, the color quantization techniques use only the color values and not any spatial information related to the structure and the texture of the images.

Multithresholding is a technique in which by using only the image histograms, proper threshold values are determined to define the limits of the image color classes, the centers of which correspond to the final image colors. Mainly, these techniques have been used for gray-scale images [16], [20], [22], [27]. Their extension to the color image case requires their application to the three color components histograms independently. The application of a multithresholding technique is based on the assumption that object and background pixels in a digital image can be well distinguished by their color values [12]. Therefore, in complex images, such as nature, texture or badly illuminated images, the multithresholding techniques do not give satisfactory results. Also, in multiobject images there are several difficulties concerning the multilevel threshold selection associated with the color distributions and the existence of small objects and object overlapping. It should be noticed that a multithresholding technique does not take into account the local texture characteristics of the images.

This paper proposes a new adaptive color reduction technique, which not only exploits the colors of the images, but also their local spatial characteristics. The proposed approach significantly improves previous techniques [17]–[19] by introducing

Manuscript received November 13, 2000; revised September 9, 2001. This paper was recommended by Associate Editor V. Govindaraj.

N. Papamarkos and C. Strouthopoulos are with the Electric Circuits Analysis Laboratory, Department of Electrical and Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece (e-mail: papamark@ee.duth.gr).

A. E. Atsalakis is with the Electric Circuits Analysis Laboratory, Department of Electrical and Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece. He is also with the Cultural and Educational Technology Institute, 67100 Xanthi, Greece.

Publisher Item Identifier S 1083-4419(02)00122-X.

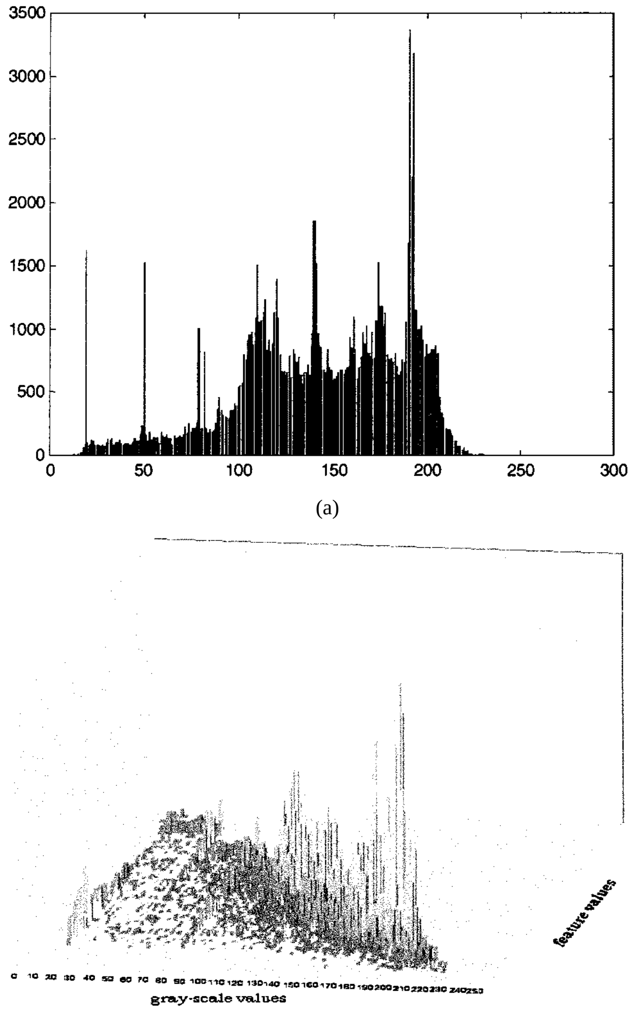


Fig. 1. Influence in the feature space of a simple spatial feature. (a) Image histogram. (b) Transformation of the feature space if a max spatial feature is applied.

an adaptive clustering scheme, which is based on a tree decomposition procedure suitable for color images. The proposed technique can be used with any color scheme and is suitable for reduction of the colors of the images to a small number. In addition, class split and merging conditions are introduced in order to define if a feature class must be further split or merged. The color values of each pixel are related to local spatial features extracted from its neighboring region. Thus, the three-dimensional (3-D) histogram clustering approach of the color multithresholding techniques is extended to a multidimensional feature clustering technique. Fig. 1 clarifies the effect on the color space when the max operator is applied, as an additional feature, to the red component of the peps image [Fig. 9(a)]. As it can be observed, the feature space became two-dimensional (2-D) with increased variance and, therefore, with better clustering capabilities.

The RGB (or any other color scheme) color components of each pixel are considered as the first three features. The entire feature set is completed by additional spatial features which are extracted from neighboring pixels. These features are associated with spatial image characteristics, such as min, max, entropy values, etc. The structure of the new color reduction technique allows the use of a different feature set in each level of the tree. The adaptive clustering scheme has the form of a tree with levels and nodes. Fig. 2 depicts a simple form of a tree that can be used for the adaptive color reduction procedure. According to this scheme, the color reduction technique is applied level-by-level, starting from the root node until all the tree nodes are examined. In each node, the features (colors and spatial features) of the node are classified in m classes. Then, the pixels of each one of the m classes are further classified by the color reduction technique, using a different spatial feature set. The split and merging conditions define if the classes can be split or merged. Adjusting these conditions, the color reduction algorithm can be used as an estimator of the number of the dominant colors in the images.

In each node of the tree, the feature sets feed a self-organized NNC, which consists of a PCA and a Kohonen SOFM [7], [13]. The PCA is used to manipulate the coordinate axes that the data falls on. The new axes are uncorrelated and they represent the maximum variability that occurs in the process data. The SOFM is competitively trained according to the Kohonen's learning algorithm. After training, the output neurons of the SOFM define the proper feature classes. Next, each pixel is classified into one of these classes and resumes the color of the class. In that way, the original image is converted into a new one, which has a limited number of colors and its spatial characteristics approximate those defined by the features used. In order to reduce storage requirements and cut off computation time, the training set must be a representative sample of the image pixels. In our approach, the subsampling is performed via a fractal scanning technique based on Hilbert's space filling curve [21]. The Hilbert curve has been used in many applications. Recently, some image compression methods based on the Hilbert curve were developed [11]. The Hilbert curve is used in adaptive color reduction (ACR) only for data subsampling (to speed up the entire process) and appropriate feature extraction. The important feature of the Hilbert curve is that it continuously scans the neighboring entry in the image [4].

The proposed method was tested in a variety of images and the results are compared to other color reduction techniques. This paper presents characteristic examples of the application of the proposed technique to images using various parameters. The entire technique has been implemented in a visual environment using C++ and the program can be downloaded from the site: http://ipml.ee.duth.gr/~papamark/Index.html.

II. Description of the Method

A color image could be considered as a set of $n \times m$ pixels where the color of each pixel is a point in the color space. There are many color spaces used for color images. However, the proposed method can be applied to any type of color space. A color space can be considered as a 3-D vector space where each pixel (i, j) is associated with an ordered triple of color components $(c_1(i, j), c_2(i, j), c_3(i, j))$. Therefore, a general color image function can be defined by the relation

$$I_k(i, j) = \begin{cases} c_1(i, j), & \text{if } k = 1 \\ c_2(i, j), & \text{if } k = 2 \\ c_3(i, j), & \text{if } k = 3. \end{cases} \tag{1}$$

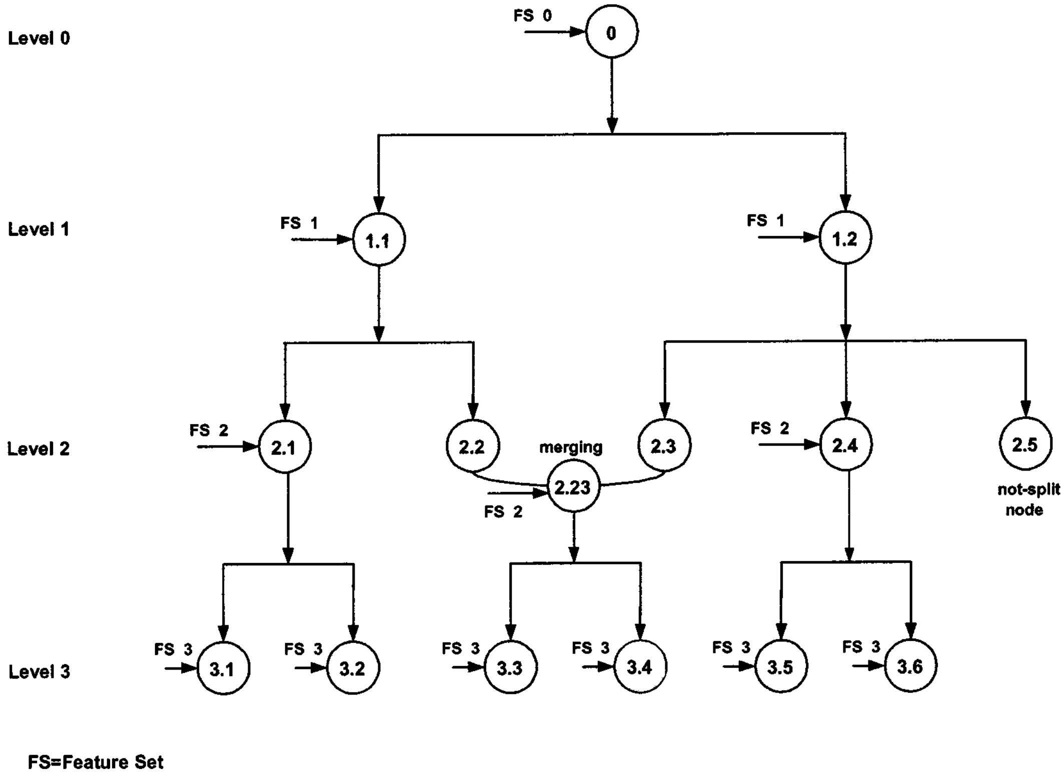


Fig. 2. Tree scheme of the adaptive color reduction algorithm.

Each primary color component represents an intensity which varies from zero to a maximum value. For example, in the RGB space, color allocation is defined over a domain consisting of a cube, with opposite vertices at (0, 0, 0) and $(C_{r\_\max}, C_{g\_\max}, C_{b\_\max})$. Usually, $C_{r\_\max} = C_{g\_\max} = C_{b\_\max} = 255$.

Let $N(i, j)$ denote the local neighboring region of pixel (i, j). Usually, $N(i, j)$ is considered to be a 3×3 or a 5×5 mask where the pixel (i, j) is the center pixel of the mask. We assume that every pixel (i, j) belongs to its neighborhood $N(i, j)$. It is obvious that in most cases, the color of each pixel is associated with the colors of the neighboring pixels and the local texture of the image. Therefore, the color of each pixel (i, j) can be associated with local image characteristics extracted from the region $N(i, j)$. These characteristics can be considered as local spatial features of the image and can be helpful for the color reduction process. That is, using the values of the colors of $N(i, j)$, f_k, $k = 4, 5, \ldots, K+3$ local features can be defined which are considered next as image spatial features. As it can be observed in Fig. 3, each pixel (i, j) is related to its $(c_1(i, j), c_2(i, j), c_3(i, j))$ color values which are considered to be the first spatial features with K additional features. This approach transforms the 3-D color feature space of the classical color quantization techniques to a more advantageous one of $K+3$ dimensions. No restrictions are implied for the type of local features. However, the features must represent simple spatial characteristics, such as min, max, entropy, and median values of the neighboring masks, etc. Fig. 1 presents an example of how a simple max feature influences and transforms the initial Red component (in an RGB color scheme) of the peps image [Fig. 9(a)]. The feature space became 2-D and the feature classes can now be obtained more easily.

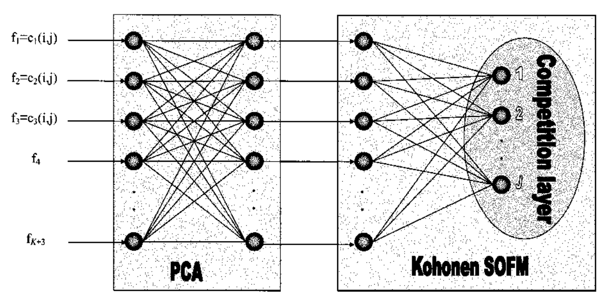


Fig. 3. Structure of the NNC.

According to the above analysis, the color reduction problem can be considered as the problem of best transforming the original color image to a new one, having only J colors, so that the final image approximates not only the principal color values, but also the local characteristics used. An effective approach to solve this problem is to consider it as a clustering problem and achieve its solution using suitable classifiers. It is obvious that every time that we use additional features, the feature space increases and color classes that could not be split before can now be separated. This is useful in many cases such as in color documents where text and background colors are similar. To solve this problem, the color reduction procedure must be applied in an adaptive mode. Specifically, this procedure follows a tree structure (Fig. 2) with levels and nodes. In each level, an additional and proper set of features can be used so that new

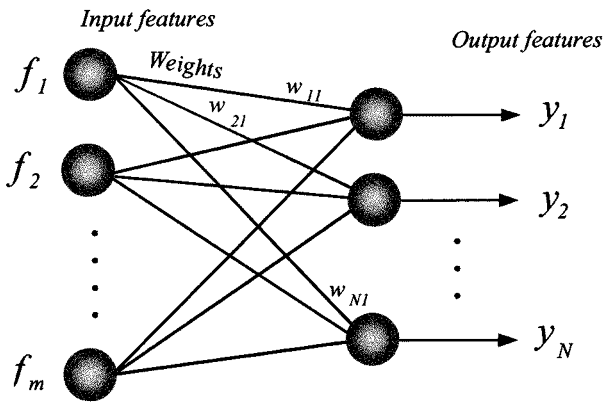


Fig. 4. PCA NN.

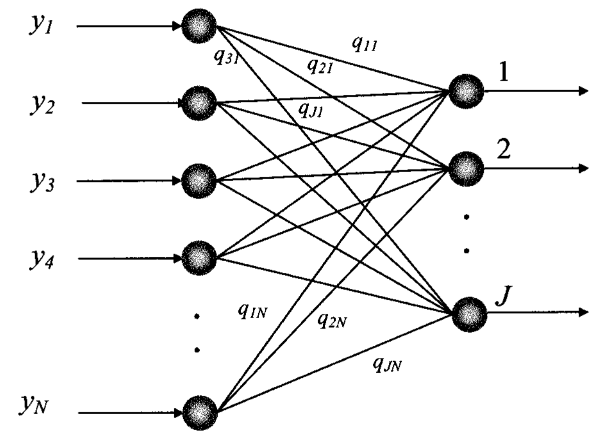


Fig. 5. SOFM NN.

classes become visible. In each tree node, the color reduction technique is performed only on the pixels of the initial image that correspond to the color class produced in the previous level. The entire procedure is terminated when all the nodes of the tree have been examined. In the final stage, the extracted color image components are merged by a simple merging procedure.

The classifier used in each tree node is a powerful self-organized NNC with the structure shown in Fig. 3. As it can be observed, it consists of a PCA and an SOFM NN. The use of PCA is essential due to the multidimensionality of the feature space. Through the PCA transform, the maximum variance in the input feature space is achieved and, hence, the discrimination ability of the SOFM is increased. It is well known that the main goal of an SOFM NN is the representation of a large set of input vectors with a smaller set of "prototype" vectors, so that a "good" approximation of the original input space can be succeeded. In other words, an SOFM NN optimally decreases the input feature space into a smaller one. The resultant feature space can be viewed as a representative of the original feature space and, therefore, it approximates statistical characteristics of the input space.

The PCA has $K+3$ input and N output neurons. Usually, N is taken equal to $K+3$. It is used to increase the discrimination of the feature space. The SOFM has N input and J output neurons. The entire NN is fed by the extracted features and, after training, the neurons in the output competition layer of the SOFM define the J classes. Using the NN, each image pixel is classified into one of the J classes and it converts its color to the color defined by the class.

A. PCA Neural Network (NN)

As it was mentioned above, the first stage of the NN classifier is a PCA. The PCA NN transforms the input feature space to a new one in order for the maximum discrimination in the feature space to be obtained. The transformation is designed in such a way that the original feature set is represented by a number of effective "features" and yet retains most of intrinsic information contained in the data. Here, a single-layer feedforward NN is used to perform a PCA. Its structure is given in Fig. 4. The PCA is trained using the Generalized Hebbian Algorithm (GHA) [7]. This is an unsupervised learning algorithm based on a Hebbian learning rule. The Hebbian rule states that if two neurons on either side of a synapse are activated simultaneously, the strength of that synapse is selectively increased, otherwise it is weakened or eliminated. The GHA is implemented in the training stage via the relations

$$\Delta w_{j,i}(n) = \eta y_j(n) f_i(n) - \eta y_j(n) \sum_{k=0}^{j} w_{k,i}(n) y_k(n) \quad (2)$$

$$y_j(n) = \sum_{i=1}^{m} w_{j,i}(n) f_i(n) \quad \text{with} \quad i = 1, \ldots, m \quad \text{and} \quad j = 1, \ldots, N \quad (3)$$

$$w_{j,i}(n+1) = w_{j,i}(n) + \Delta w_{j,i}(n). \quad (4)$$

In our approach, the input vector is the $K+3$ dimensional feature vector $\boldsymbol{F}$ and the output is usually taken equal to $N = K+3$ dimensional vector $\boldsymbol{Y}$ which is computed by the relation

$$\boldsymbol{Y} = \boldsymbol{wF} \quad (5)$$

where $\boldsymbol{w}$ is the matrix of the PCA coefficients.

It should be noticed that after training, $\boldsymbol{w}$ approximates with probability one the matrix whose rows are the eigenvectors of the covariance matrix, formed by decreasing eigenvalues. The output values of the PCA are the projections of the input vector to these eigenvectors.

B. SOFM NN

The main goal of the Kohonen SOFM NN is to classify a set of vectors into a one-dimensional (1-D) or 2-D matrix. It transforms the input of an arbitrary dimension into a 1-D or 2-D discrete map subject to a topological (neighborhood preserving) constraint. The structure of the Kohonen SOFM NN used is depicted in Fig. 5. It has N input and J output neurons arranged in a 1-D grid. The input neurons are fed with the output values of the PCA. Each neuron in the competition layer represents one class which is associated not only with the color values but also with the spatial features used. By adjusting the number of the output neurons, we can define the number of the colors in the final image. The output J neurons are related to the input neurons via the $q_{j,i}$, $i = 1, \ldots, N$ and $j = 1, \ldots, J$ coefficients.

The SOFM is competitively trained according to the following weight update function

$$\Delta q_{ji} = \begin{cases} a(y_i - q_{ji}), & \text{if } |c-j| \le d \\ & \text{for } i = 1, \ldots, N \quad \text{and} \\ & j = 1, \ldots, J \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where y_j are the input values, a the learning parameter, d the neighboring parameter (a typical initial value is less than 0.25), and c the winner neuron. The values of parameters a and d are reduced to zero during the learning process. This learning algorithm is referred to as Kohonen's learning.

After training, the optimal color values correspond to the centers of the features classes obtained by the NN. Specifically, the values of the three color components for each optimal color are equal to the first three elements of the vectors

$$\boldsymbol{c} = \boldsymbol{w}^T \boldsymbol{q}^{(j)} \qquad j = 1, \ldots, J. \tag{7}$$

Next, the original image is rescanned and by using the NN, the new image is constructed with only J color values in such a way that approximates the texture of the original image, according to the spatial features used. This procedure is applied to each tree node and the final image is constructed by simply merging the color images obtained in the nodes.

C. Split and Merging Conditions

To avoid the split of compact classes, split conditions can be used during the adaptive clustering process. The split conditions define when a class must be split more. These conditions are applied locally to each tree node or to each level of the tree. Analytically, the split conditions used are as follows.

a) *Variance of each class*: A class is not split further if its variance is less than a threshold value.

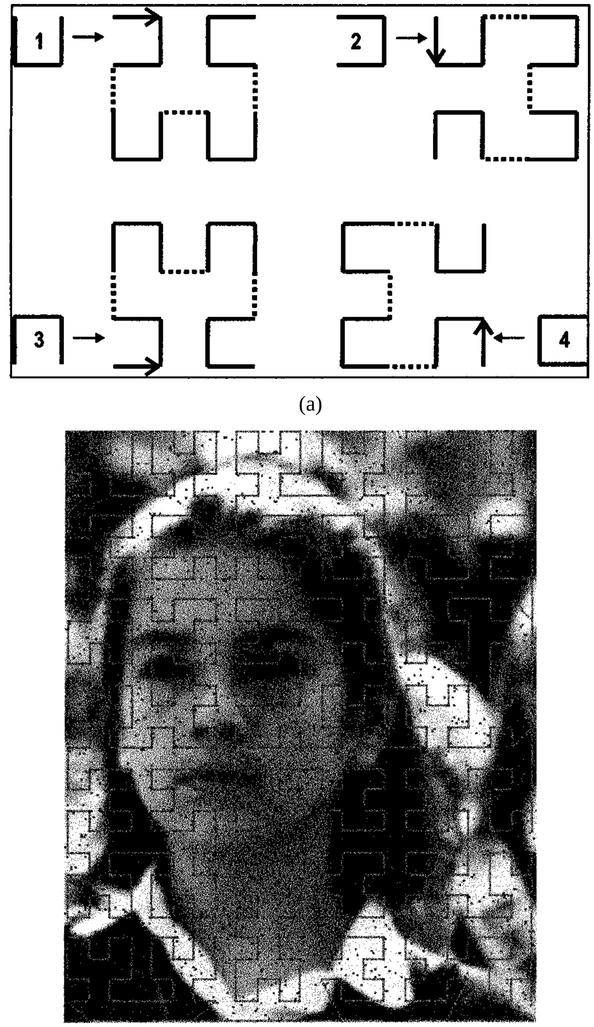
b) *Variance of the class centers*: The classes produced in each node of the tree are not split further if the variance of their centers is less than a threshold value.

c) *Increasing of the variance*: In each level of the tree, each class is accepted if its appearances increase the variance of the centers of the classes. Otherwise, the specific class will not be split further.

d) *Minimum number of pixels in each class*: The minimum number of pixels classified in each class cannot be less than a threshold value.

e) *Minimum number of samples*: In each node of the tree, the number of the training points, i.e., the number of pixels belonging to the node class must exceed an upper bound.

Additionally, in order to increase the clustering capabilities, merging conditions can be used in each level of the tree defining when classes are close enough and must be merged. Specifically, in each level of the tree, the Euclidean distances between the classes are determined and then the classes that have distances less than a threshold value, expressed as a percentage of the mean distance, are merged. This procedure is important because it leads to good color segmentation results by merging color classes that come from different paths of the tree.



(a)

(b)

Fig. 6. (a) Hilbert's curve constructing rule. (b) Fractal subsampling.

D. Image Subsampling

The proposed technique can be applied to color images without any subsampling. However, in the case of large-sized images and in order to achieve reduction of the computational time and memory size requirements, it is preferable to have a subsampling version of the original image. In order for the subsampling image to be a better representation (to capture better the image texture) of the original one, we choose to use a fractal scanning process, based on the well-known Hilbert's space filling curve [21]. The technique of fractal scanning technique is much more efficient in capturing features in a digital image than the commonly used techniques of raster scanning. The raster scanning techniques do not maintain the adjacency of the features in the image since their scanning direction is usually not identified with the one that the features specify. Feature extraction is more substantial when we succeed to retain the neighborhood relationship between the pixels. Thus, it is desirable that a technique scans one area of the image completely before moving to the next.

To generate the Hilbert curve we must start with the basic staple-like shape as depicted in Fig. 6(a). The rest of the Hilbert curve is created sequentially using the same algorithm. Starting

(a)

(b)

(c)

Fig. 7. (a) Original image with 179 473 unique colors. (b) Document with only four colors. (c) Document with four unique colors obtained using split and merging conditions.

with the basic Hilbert curve, in each step we increase the grid size by a factor of two. Then, we place four copies of the previous curve on the grid. The lower two copies are placed directly as they are. The upper left quarter is rotated by 90° counterclockwise and the upper right quarter by 90° clockwise. Finally, the four pieces are connected with short straight segments to obtain the next step curve.

To further improve the subsampling process, samples are taken, using a random process, not only on the peaks of the fractal curve but also in the neighbor of the peaks. Thus, we can better adjust the number of samples and capture the local image characteristics. Also, for better training results, in each epoch of the training procedure, the samples are different and taken using a clockwise procedure from the random samples of the fractal peaks. An example of the subsampling process is given in Fig. 6(b). A complete description of Hilbert's curve is given in [21].

E. Stages of the Method

According to the above analysis, the stages of the proposed adaptive color reduction technique are the following.

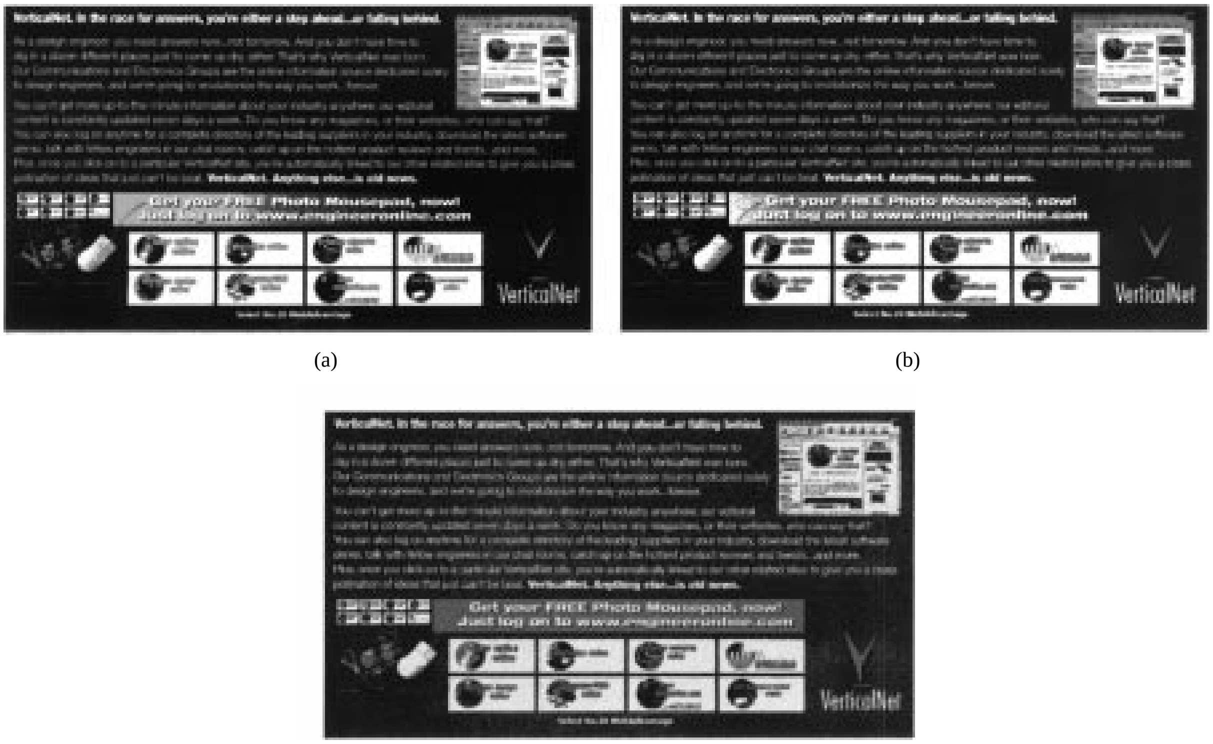
- Stage 1) Define the desired maximum number J of the final colors.
- Stage 2) Define the type of spatial features that will be used in each level of the tree.
- Stage 3) Define the split and merging conditions.
- Stage 4) Construct the PCA and SOFM NNs.
- Stage 5) Define the subsampling parameters.
- Stage 6) In each tree node, train the NN with the predefined features and then using the NN, transform the colors of the original image to those obtained.
- Stage 7) At the end of the adaptive process, construct the final image by merging the separated color regions obtained.

III. Experimental Results

The proposed method has been implemented in a visual environment. The PCA and the SOFM NNs are implemented and tested using the developer version 3.02 of the NN package NeuralSolution (NeuroDimension, Inc.). Due to the space limitation, only six characteristic experimental results are presented. However, the program implementing the entire adaptive color reduction technique can be downloaded and tested from the site: http://ipml.ee.duth.gr/~papamark/Index.html.

Experiment 1: This example demonstrates the application of the proposed method for the color reduction of the mixed-type color document shown in Fig. 7(a). The size of the original image is 1104×616 pixels with 150 dpi resolution and the number of unique colors in the image is 179 473. For subsampling, 115 280 pixels are used, distributed in ten pixels per fractal curve peak. In this example, the proposed method uses only one spatial feature described by the horizontal edge extraction mask of Prewitt

$$\begin{aligned} f_{3+k}(i,\, j) = \tfrac{1}{3}(&I_k(i-1,\, j+1) + I_k(i,\, j+1) \\ &+ I_k(i+1,\, j+1) - I_k(i-1,\, j-1) \\ &+ I_k(i,\, j-1) + I_k(i+1,\, j-1)) \\ &k = 1,\, 2,\, 3. \end{aligned} \tag{8}$$



(a) (b) (c)

Fig. 8. Color reduction using (a) MC, (b) Dekker's method, and (c) Paint Shop Pro 5 image processing software.

The form of the tree is binary with two levels each, one of which has two nodes. No split and merging conditions have been applied and the RGB color scheme is used.

The application of the proposed technique results to the image shown in Fig. 7(b) with only the following four RGB colors

R	G	B
7	21	101
114	65	84
232	230	237
186	133	105

The proposed adaptive color reduction technique can be used to determine the proper number of the dominant colors in an image. This can be done if we define suitable split and merging conditions. For example, let us consider a tree with four levels each, one with four nodes (a total number of 256 nodes) and the following split and merging conditions.

- The split threshold value for the variance of each class is taken equal to seven.
- The threshold value for the variance of class centers is equal to 12.
- The minimum number of samples is equal to 20.
- The minimum number of pixels in each class must be greater than 100.
- The increasing of the variance condition.
- The threshold value for the merging condition is taken equal to 70% of the mean distance.

Using these parameters, the adaptive color reduction technique results only to the following four colors

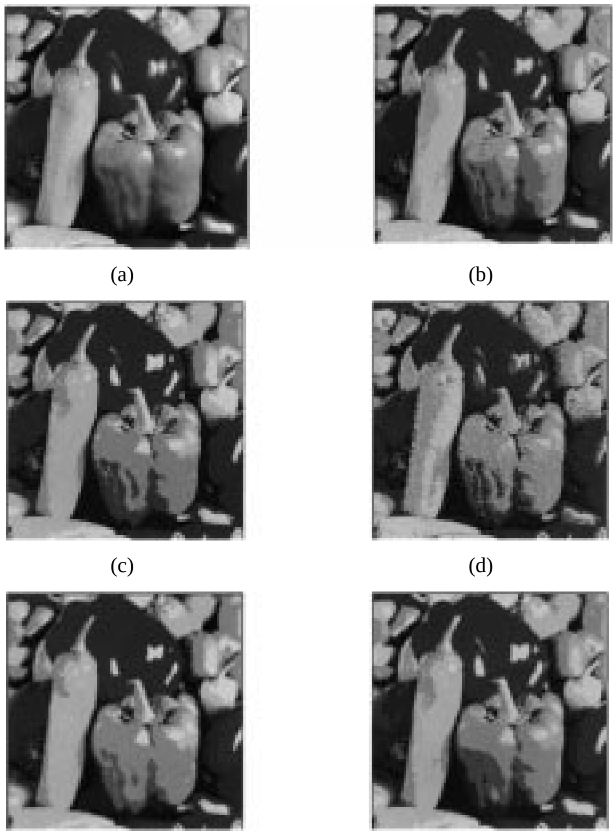
R	G	B
7	21	102
178	108	81
228	225	232
28	39	102

and the image obtained is depicted in Fig. 7(c). Comparing Fig. 7(b) and (c), we can observe that the two images have similar colors.

For comparison reasons, the colors of the same image are reduced to four using the method of Dekker (NQ)[5], MC [8], and the Paint Shop Pro image processing software. The images obtained are shown in Fig. 8(a)–(c), respectively. As it can be optically observed, the proposed technique has better color segmentation results. The MC algorithm provides good color results but, unfortunately, the color area produced is not solid and many regions with dithering appear.

It should be noticed that the results obtained using the proposed technique are not unique. Different color reduction results can be obtained if the system is fed with different spatial features, split and merging conditions, etc.

Experiment 2: This second example demonstrates the influence of different types of spatial features applied on the image of Fig. 9(a). The image size is 200 $\times$ 200 pixels with 94 dpi resolution and 26 422 unique colors. The subsampling starts



(a) (b) (c) (d) (e) (f)

Fig. 9. (a) Original image, (b) image with nine colors without using spatial features, and images with nine colors obtained by using the (c) contour, (d) Kirsch, (e) min, and (f) max operators.

with 18 200 samples with seven random samples per fractal peak. The color reduction algorithm is applied using the RGB color scheme and no split and merging conditions are specified. Fig. 9(b) indicates the results obtained for nine colors and without using spatial features. In this example, four different types of spatial features are applied independently. The first feature is the contour feature

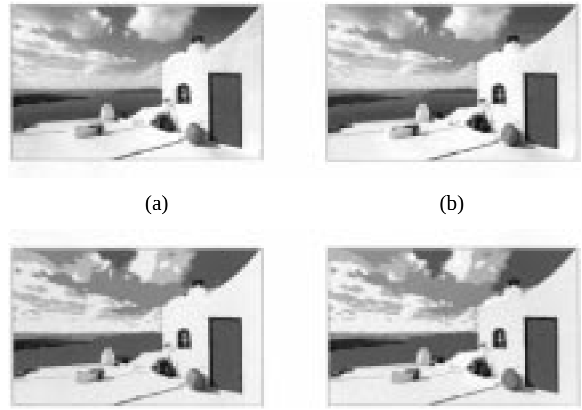
$$f_{3+k}(i, j) = 8I_k(i, j) - \sum_{a=-1}^{1} \sum_{\substack{b=-1 \\ (a,b)\neq(0,0)}}^{1} I_k(i+a, j+b) \qquad k = 1, 2, 3 \text{ with bias} = 255. \tag{9}$$

The second feature is extracted by using the Kirsch mask [25]

$$\begin{aligned} f_{3+k}(i, j) = 3\,[&I_k(i-1, j-1) \\ &+ I_k(i, j-1) + I_k(i+1, j-1) \\ &+ I_k(i+1, j) + I_k(i+1, j+1)] \\ &- 5\,[I_k(i-1, j) + I_k(i-1, j-1) \\ &+ I_k(i, j+1)] \qquad k = 1, 2, 3. \end{aligned} \tag{10}$$

The Kirsch operators are edge detection operators and approximate the image first derivative. The other two features are extracted by using the nonlinear operation of min and max. The min operator applies a spreading of black areas and shrinking of white areas. The max operator has the effect of applying a spreading out of white areas and choking in black areas. In order to clarify the influence of the feature type, Fig. 9 shows the images obtained by the application of the contrast, Kirsch, min, and max operators. We can see that the texture of the images obtained approaches the characteristics of the spatial features used.

Experiment 3: This experiment demonstrates the application of the proposed method to the natural image of Fig. 10(a), whose size is 613 $\times$ 384 pixels with 300 dpi and has 54 875 unique colors. The proposed technique is applied for a tree constructed of three levels with six, three, and two nodes per level, respectively. This structure means that the tree has $6 \times 3 \times 2 = 36$ nodes. In the first two levels the contour features (9) are used



(a) (b) (c) (d)

Fig. 10. (a) Original image. (b) Image with 32 unique colors. (c) Image with only 11 unique colors. (d) Image with 14 colors obtained using HSI color space.

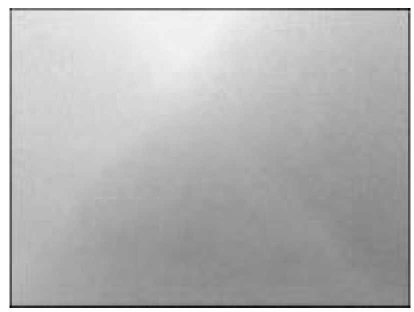


Fig. 11. Original image.

TABLE I
QUANTIZED IMAGES

Colors	2	4	8	16	32	64	128	256
ACR								
MC								
Padie								
Wu								
NQ								
MV								

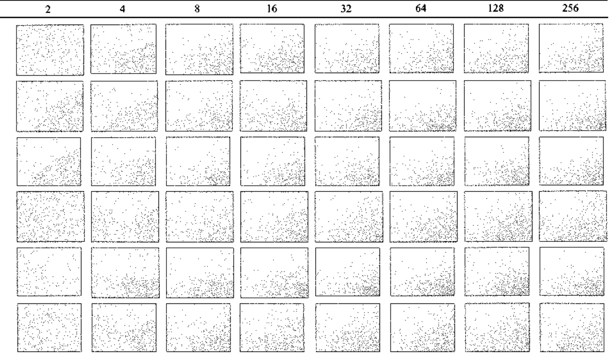


TABLE II
COMPARATIVE QUANTIZATION ERROR RESULTS

	2-colors						4-colors					
	MAD	NED	AQE	ADC	SNR	PSNR	MAD	NED	AQE	ADC	SNR	PSNR
ACR	148	0.257	64.763	60.665	14.247	15.757	84	0.167	43.409	39.781	17.978	19.559
MC	158	0.303	77.427	61.176	12.828	14.640	124	0.185	46.102	41.097	17.910	18.701
Padie	173	0.295	74.324	61.125	13.046	14.627	104	0.192	48.728	40.848	16.767	18.348
Wu	164	0.257	65.125	60.304	14.245	15.827	95	0.171	44.605	40.848	17.769	19.350
NQ	201	0.280	66.897	69.695	13.505	15.086	94	0.172	44.313	44.358	17.736	19.317
MV	165	0.257	64.794	60.389	14.164	15.845	95	0.171	44.181	40.828	17.814	19.395
	8-colors						16-colors					
ACR	63	0.102	25.812	29.838	22.223	23.804	47	0.065	15.967	21.591	26.178	27.759
MC	79	0.121	30.986	28.845	20.803	21.809	48	0.084	21.248	20.566	23.946	25.887
Padie	86	0.115	28.407	31.838	21.245	22.826	66	0.075	17.893	22.794	24.963	26.544
Wu	72	0.109	27.047	31.397	21.697	23.277	48	0.072	18.004	21.828	25.329	26.910
NQ	78	0.104	25.998	31.563	22.074	23.655	57	0.068	16.048	22.722	25.864	27.444
MV	82	0.110	26.975	32.693	21.624	23.205	55	0.071	17.156	22.273	25.458	27.039
	32-colors						64-colors					
ACR	34	0.043	10.090	15.846	29.716	31.286	25	0.029	6.464	11.343	33.195	34.729
MC	36	0.052	12.273	15.606	28.145	29.820	27	0.033	7.697	11.475	31.955	32.708
Padie	55	0.056	11.586	17.343	28.219	29.799	33	0.032	7.138	11.883	32.297	33.878
Wu	39	0.049	11.987	16.582	28.644	28.644	28	0.035	8.763	11.854	31.508	33.089
NQ	41	0.046	10.584	16.713	29.190	30.770	40	0.035	7.732	13.475	31.471	33.051
MV	35	0.047	11.131	15.968	29.055	30.636	32	0.030	7.038	10.921	32.830	34.410
	128-colors						256-colors					
ACR	16	0.020	4.297	7.622	36.515	38.096	13	0.014	3.072	5.461	39.269	40.852
MC	22	0.024	5.583	8.438	34.793	35.463	16	0.017	3.886	5.850	37.974	37.890
Padie	25	0.021	4.432	8.348	36.008	37.589	18	0.015	2.823	5.770	39.520	41.100
Wu	20	0.027	6.936	9.268	33.673	35.254	15	0.023	6.131	7.284	35.052	36.633
NQ	29	0.025	5.377	9.573	34.547	36.128	31	0.024	4.782	9.169	34.915	36.496
MV	21	0.021	4.688	7.824	36.056	37.637	15	0.015	3.197	5.545	39.308	40.888

while, in the last level, we apply features extracted by using the mean value operator. The sampling points in each node are equal to 3981 $\times$ 10 (ten random samples per fractal peak) and the RGB color system is used. Initially, the algorithm runs without applying any split or merging condition and results in the image shown in Fig. 10(b). Next, we solve the same problem using now split and merging conditions. Specifically, the threshold value for the merging conditions is taken equal to 20% while for the split conditions we use only the variance of the class centers condition. Doing this, the color reduction algorithm results in the image of Fig. 10(c), which has only 11 unique colors. Comparing Fig. 10(b) and (c), it can be observed that in the resultant image, the dominant colors remain. If we solve the same problem using the HIS color space, the technique results in the image shown in Fig. 10(d) which has only 14 colors. Even if the image has a larger number of colors, the colors obtained are close to those of Fig. 10(c).

Experiment 4: In this experiment we present comparative results between the proposed ACR algorithm and other well-known color quantization techniques. It should be noticed that the proposed technique could not be considered only as a color quantization method due to its ability to exploit spatial characteristics in an adaptive process. That is, the comparative results presented in this example are only referred to as the basic mode of operation of the ACR technique and does not clarify its advantages in other tasks. Some of these advantages are clarified in other experiments.

One of the most important goals of a quantization process is to make the *perceived difference* between the original and its quantized representation as small as possible. The perceived difference can be considered also as *perceived image quality or perceived image similarity*. In general it is very difficult to define and calculate objectively such measures. In fact, there is no good objective criterion available. However, in image quantization literature it is common to use some image dependent distortion measures [2], [8], [26], [29]. The comparative results presented here are based on six criteria. If we define as $A(i, j, k)$, $i = 1, \ldots, N$ and $j = 1, \ldots, M$ the original image and $B(i, j, k)$ the corresponding quantized one, then the criteria are defined as

a) Maximum Absolute Difference (MAD):

$$MAD = \text{maximum}|A(i, j, k) - B(i, j, k)|$$
$$i = 1, \ldots, N. \quad j = 1, \ldots, M \quad \text{and} \quad k = 1, \ldots, 3. \tag{11}$$

b) Average Quantization Error (AQE):

$$AQE = \frac{1}{N \cdot M} \sum_{i=1}^{N} \sum_{j=1}^{M} \sqrt{\sum_{k=1}^{3} (A(i, j, k) - B(i, j, k))^2}. \tag{12}$$

c) Normalized Euclidean Distance (NED):

$$NED = \frac{1}{N \cdot M} \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{M} \left[\sum_{k=1}^{3} (A(i, j, k) - B(i, j, k))^2 \right]}. \tag{13}$$

d) Average Distortion Per Color (ADC):

$$ADC = \frac{1}{N_c} \sum_{i=1}^{N_c} ||C_i - q(C_i)|| \tag{14}$$

where C is the set of original image unique colors, N_c the number of unique colors, and $q(x)$ the corresponding quantized color of color x.

e) Signal-to-Noise Ratio (SNR): This function calculates the SNR in decibels (dB). In our case, it is defined by the ratio of the average power in the original image A to the average power of the noise image. The noise image is obtained by subtracting the original image A from the quantized image B.

$$SNR = 10\log_{10}\left(\frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\left[\sum_{k=1}^{3} A^2(i,j,k)\right]}{\sum_{i=1}^{N}\sum_{j=1}^{M}\left[\sum_{k=1}^{3}(A(i,j,k)-B(i,j,k))^2\right]}\right). \quad (15)$$

f) Peak Signal-to-Noise Ratio (PSNR): The PSNR for images is defined in (16) as shown at the bottom of the page. It is noted that the denominator in the above equation is equal to the mean squared error (MSE) between A and B images.

The results of the adaptive color reduction algorithm presented in this experiment are obtained without use of the PCA and with the following default specifications for the Kohonen SOFM.

- Linear reduction of step size and neighborhood.
- Step size beta $= -7.916\,67\text{e} - 5$.
- Initial neighborhood = round [0.6 $\times$ output neurons].
- Neighborhood beta $= -0.016$ when output neurons $= 3$.
- SOFM weight variance $= 0.000\,001$.
- Max epochs = 5000.

For comparison reasons, we use the methods of NQ [5], MC [8], Padie [10], minimum variance (MV) [15], [30], and Wu [31]. All techniques are applied to the image shown in Fig. 11, which has 321 $\times$ 240 pixels and 22 350 unique colors. Table I demonstrates the quantization results obtained for two, four, eight, 16, 32, 64, 128, and 256 colors. Table II gives the values for the six quantization criteria and Fig. 12 presents graphically the variation of some of them. These results clearly indicate that the ACR provides, in most of the cases, and according to the majority of the criteria, the best quantization results. However, again it should be noticed that the ACR technique could be used in many different ways and not only for color quantization.

Experiment 5: In this example, the ACR technique is applied to the complex color document shown in Fig. 13(a). This document has a large-size noisy background, text, and a color image. The original image has 1275 $\times$ 1755 pixels and 133.844 unique colors. The application to this document of any color quantization technique leads to images with noisy background. This happens because the basic criterion of the color quantization technique is the color similarity for any pixel. Fig. 13(b)–(e) give

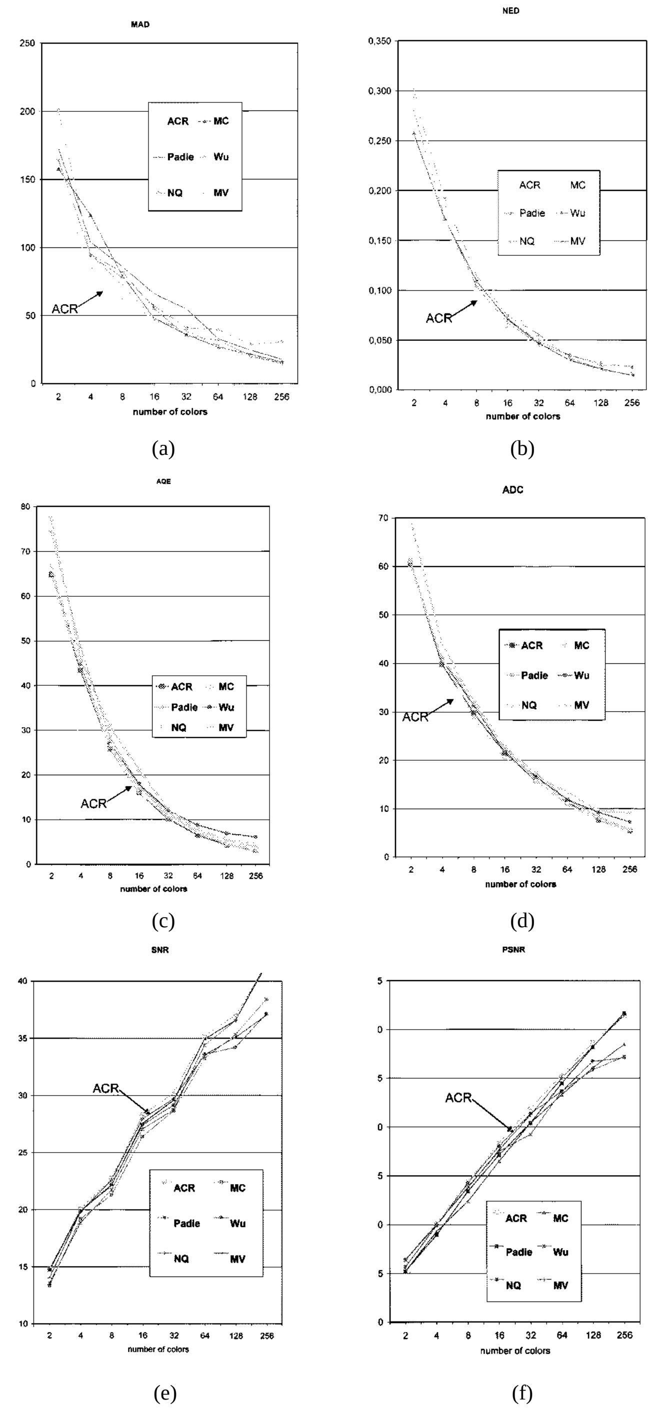
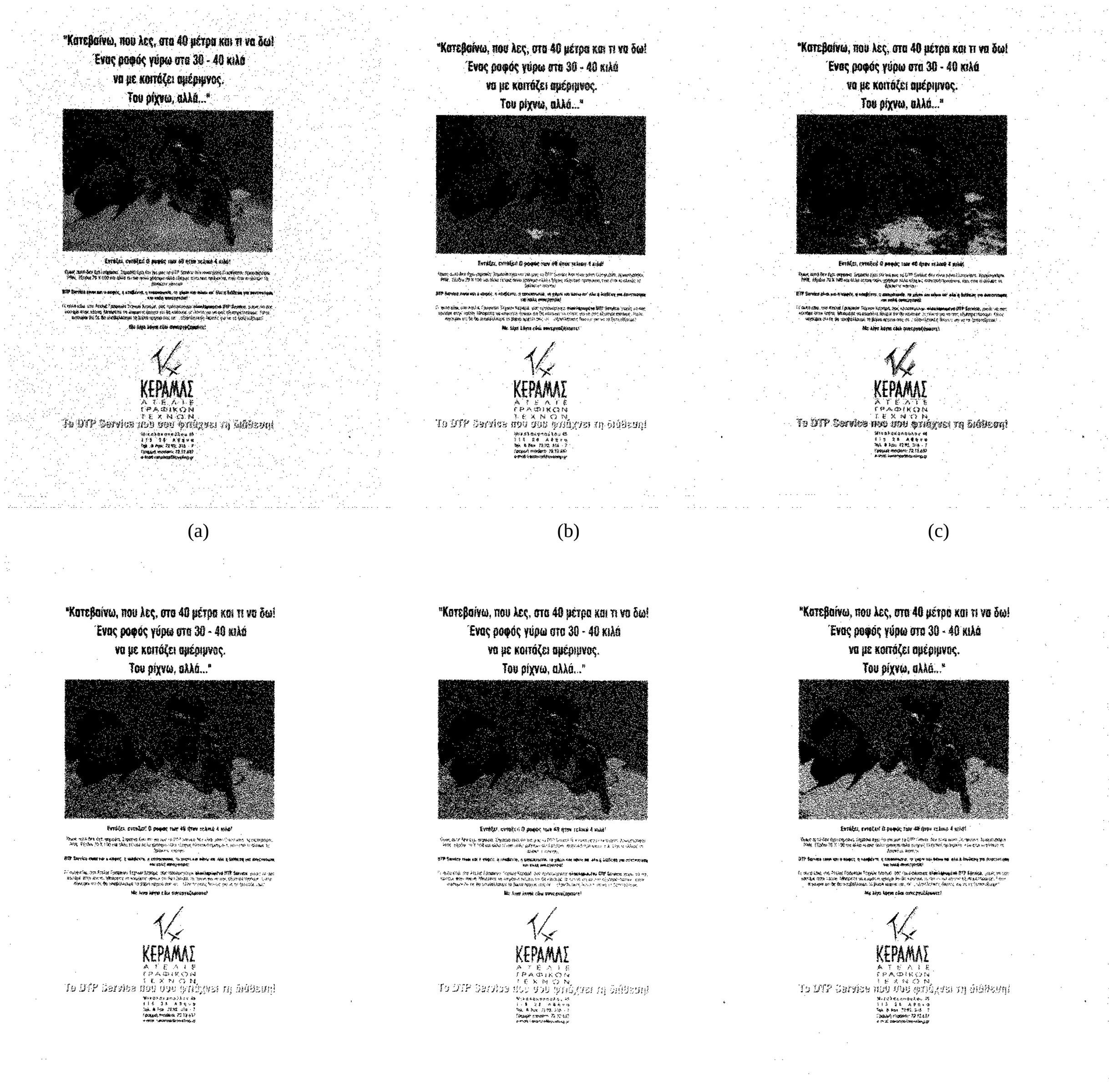


Fig. 12. Graphical representation of quantization errors.

$$PSNR = 10\log_{10}\left(\frac{255^2}{\frac{1}{3\cdot N\cdot M}\sum_{i=1}^{N}\sum_{j=1}^{M}\left[\sum_{k=1}^{3}(A(i,j,k)-B(i,j,k))^2\right]}\right). \quad (16)$$



(a) (b) (c)

(d) (e) (f)

Fig. 13. Comparative image quantization results. (a) Original. (b) MC. (c) NQ. (d) Wu. (e) MV. (f) ACR.

TABLE III
COLOR QUANTIZATION ERRORS FOR EXPERIMENT 5

	14-colors					
	MAD	NED	AQE	ADC	SNR	PSNR
ACR	131	0.011	13.417	28.851	27.391	28.7452
MC	122	0.0158	17.949	41.523	24.073	25.427
Wu	156	0.0095	10.654	29.667	28.491	29.8457
NQ	189	0.0147	13.205	46.444	24.711	26.065
MV	230	0.0135	13.160	39.473	25.453	26.808

the quantization results obtained by using MC, NQ, Wu, and MV techniques for 14 colors. The ACR technique is applied by using the following.

- A $3 \times 3 \times 3 \times 3$ tree structure.
- 5×5 mean spatial features in every tree node.
- A threshold value of 35% for the merging conditions.

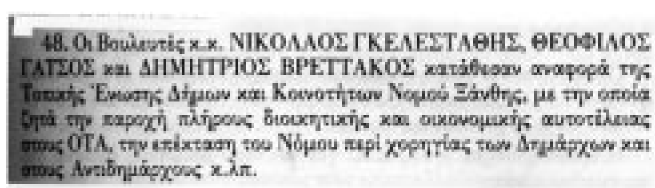

48. Οι Βουλευτές κ.κ. ΝΙΚΟΛΑΟΣ ΓΚΕΛΕΣΤΑΘΗΣ, ΘΕΟΦΙΛΟΣ ΓΑΤΣΟΣ και ΔΗΜΗΤΡΙΟΣ ΒΡΕΤΤΑΚΟΣ κατάθεσαν αναφορά της Τοπικής Ένωσης Δήμων και Κοινοτήτων Νομού Ξάνθης, με την οποία ζητά την παροχή πλήρους διοικητικής και οικονομικής αυτοτέλειας στους ΟΤΑ, την επέκταση του Νόμου περί χορηγίας των Δημάρχων και στους Αντιδημάρχους κ.λπ.

Fig. 14. Original image of Experiment 6.

The final image obtained is shown in Fig. 13(f), where it can be observed that the background noise has been substantially suppressed, i.e., the background obtains homogeneous colors and, in general, the quantized document is optically better. It should be noticed that this happens though the quantization errors, given in Table III, are not always the smallest in the ACR technique.

Experiment 6: This example demonstrates the usefulness of the ACR technique to filter, using suitable *spatial* features, in

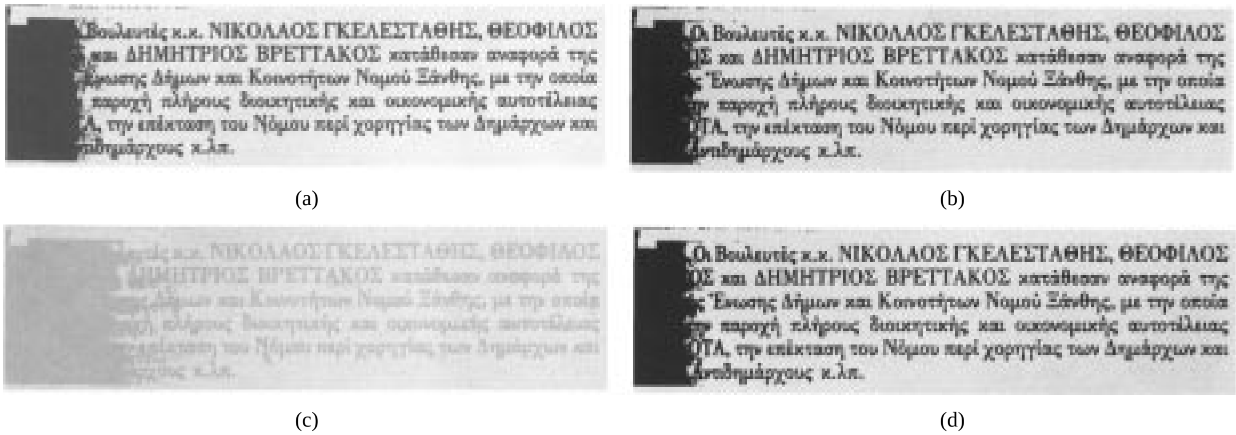


Fig. 15. Reduction of the document colors to two by using the (a) MC, (b) Wu, (c) NQ, and (d) MV techniques.

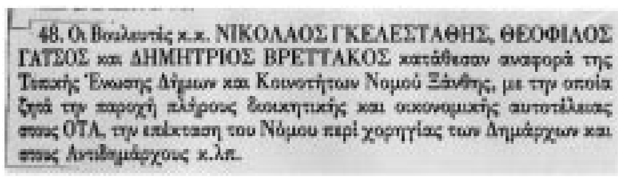


Fig. 16. Final image, with only two colors, obtained by using the ACR technique.

a badly illuminated document. The original color document is shown in Fig. 14 and our aim is to reduce the colors to two and simultaneously filter the image. The application of any quantization technique, even the ACR technique without spatial features, results into a dark zone in the left part of the document that covers the text (see Fig. 15). To overcome these difficulties the ACR technique is applied by using the following.

a) A $2 \times 2 \times 2 \times 2$ tree structure.
b) The following spatial features in every tree node.
   - The 5×5 minimum value.
   - The 3×3 Laplace operator

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

c) A threshold value of 75% for the merging conditions.

Fig. 16 shows the final image obtained with only two unique colors. The color reduction results are obviously better because the document is filtered and the characters are well defined.

IV. Conclusion

This paper proposes a general color reduction technique which is applicable to any color image. The proposed technique is based on a NN structure which consists of a PCA and a Kohonen SOFM. The training set of the NN consists of the image color values and additional spatial features extracted in the neighborhood of each pixel. These features describe image local characteristics. Therefore, the color of each pixel is related with the colors and the texture of the neighboring pixels. Therefore, the final image not only has the proper colors, but its structure approximates the local characteristics used. This is important for many applications such as segmentation and recognition systems. The proposed technique can be used with any color scheme and is suitable for drastic reduction of the colors contained in a color image.

The ACR procedure is accompanied with split and merging conditions that define if the classes must be split or merged. This procedure is important because it results in a small number of colors. In this way, it can be considered as a technique for the detection of the number of dominant colors in an image. This technique is especially significant in the case of mixed type color documents where we have text with different colors and many color images and graphics.

In order to speed up the entire algorithm, a fractal subsampling procedure based on Hilbert's space filling curve is applied.

References

[1] I. Ashdown, "Octree color quantization," in *Radiosity—A Programmer's Perspective*. New York: Wiley, 1994.

[2] J. M. Buhmann *et al.*, "Dithered color quantization," in *Proc. EUROGRAPHICS*, vol. 17, no. 3. Lisboa, Portugal, Computer Graphics Forum, 1998, pp. 219–231.

[3] Z. Chi, H. Yan, and T. Pham, *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition*. Singapore: World Scientific, 1996.

[4] K. L. Chung, Y. H. Tsai, and F. C. Hu, "Space-filling approach for fast window query on compressed images," *IEEE Trans. Image Process.*, vol. 9, pp. 2109–2116, Dec. 2000.

[5] A. H. Dekker, "Kohonen neural networks for optimal color quantization," *Network: Computat. Neural Syst.*, vol. 5, pp. 351–367, 1994.

[6] M. Gervautz and W. Purgathofer, "A simple method for color quantization: Octree quantization," in *Graphics Gems*, A. S. Glassner, Ed. New York: Academic, 1990, pp. 287–293.

[7] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: MacMillan, 1994.

[8] P. Heckbert, "Color image quantization for frame buffer display," *Comput. Graph.*, vol. 16, pp. 297–307, 1982.

[9] I. S. Hsieh and K. C. Fan, "An adaptive clustering algorithm for color quantization," *Pattern Recognit. Lett.*, vol. 21, pp. 337–346, 2000.

[10] [Online]. Available: http://hp.vector.co.jp/authors/VA013 060/software/padie/index.htm.

[11] S. Kamata, R. O. Eaxon, and E. Kawaguchi, "An implementation of the Hilbert scanning algorithm and its application to data compression," *IEICE Trans. Inform. Syst.*, vol. E79-D, no. 4, 1993.

[12] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognit.*, vol. 19, pp. 41–47, 1986.

[13] T. Kohonen, *Self-Organizing Maps*. New York: Springer-Verlag, 1997.

[14] Y. W. Lim and S. U. Lee, "On the color image segmentation algorithm based on the thresholding and the fuzzy C-means techniques," *Pattern Recognit.*, vol. 23, no. 9, pp. 935–952, 1990.

[15] Mathworks, Inc., *Matlab Image Processing Toolbox, Ver. 2.2* Natick, MA, 1999.
[16] N. Papamarkos and B. Gatos, "A new approach for multithreshold selection," *Comput. Vision Graph. Image Process.—Graph. Models Image Process.*, vol. 56, no. 5, pp. 357–370, 1994.
[17] N. Papamarkos, "Color reduction using local features and a SOFM neural network," *Int. J. Imag. Syst. Technol.*, vol. 10, no. 5, pp. 404–409, 1999.
[18] N. Papamarkos, C. Strouthopoulos, and I. Andreadis, "Multithresholding of color and color images through a neural network technique," *Image Vis. Comput.*, vol. 18, pp. 213–222, 2000.
[19] N. Papamarkos and A. Atsalakis, "Gray-level reduction using local spatial features," *Comput. Vision Image Understanding*, vol. CVIU-78, pp. 336–350, 2000.
[20] S. S. Reddi, S. F. Rudin, and H. R. Keshavan, "An optimal multiple threshold scheme for image segmentation," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-14, pp. 661–665, Apr. 1984.
[21] H. Sagan, *Space-Filling Curves*. New York: Springer-Verlag, 1994.
[22] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques," *Comput. Vision, Graph. Image Process.*, vol. 41, pp. 233–260, 1988.
[23] P. Scheunders, "A comparison of clustering algorithms applied to color image quantization," *Pattern Recognit. Lett.*, vol. 18, pp. 1379–1384, 1997.
[24] S. A. Shafer and T. Kanade, "Color vision," in *Encyclopedia of Artificial Intelligence*, S. C. Shapiro and D. Eckroth, Eds. New York: Wiley, 1987, pp. 124–131.
[25] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, 2nd ed. Pacific Grove, CA: Brooks/Cole, 1999.
[26] M. Stricker and M. Orengo, "Similarity of color images," in *Proc. SPIE Conf. Storage Retrieval for Still Images Video Databases*, vol. 2420. San Jose, CA, 1995, pp. 381–392.
[27] D. M. Tsai, "A fast thresholding selection procedure for multimodal and unimodal histograms," *Pattern Recognit. Lett.*, vol. 16, no. 6, pp. 653–666, 1995.
[28] L. Velho, J. Gomes, and M. V. R. Sobreiro, "Color image quantization by pairwise clustering," in *Proc. Tenth Brazilian Symp. Comput. Graph. Image Process.*, L. H. de Figueiredo and M. L. Netto, Eds. Campos do Jordao, Spain, 1997, pp. 203–210.
[29] O. Verevka, "The local K-means algorithm for color image quantization," M.Sc. dissertation, Univ. Alberta, Edmonton, AB, Canada, 1995.
[30] S. J. Wan, P. Prusinkiewicz, and S. K. M. Wong, "Variance based color image quantization for frame buffer display," *Color Res. Applicat.*, vol. 15, no. 1, pp. 52–58, 1990.
[31] X. Wu, "Color quantization by dynamic programming and principal analysis," *ACM Trans. Graph.*, vol. 11, no. 4, pp. 348–372, 1992.
[32] ——, "Efficient statistical computation for optimal color quantization," in *Graphics Gems II*, J. Arvo, Ed. New York: Academic, 1991.

Nikos Papamarkos (S'85–M'87) was born in Alexandroupoli, Greece, in 1956. He received the Diploma degree in electrical and mechanical engineering from the University of Thessaloniki, Thessaloniki, Greece, and the Ph.D. degree in electrical engineering from the Democritus University of Thrace, Xanthi, Greece, in 1979 and 1986, respectively.

From 1987 to 1996, he was a Lecturer and then Assistant Professor at the Democritus University of Thrace where he has been Associate Professor since 1996. From 1987 to 1992, he served as a Visiting Research Associate at the Georgia Institute of Technology, Atlanta, GA. His current research interests are in digital image processing, document image analysis, computer vision, pattern recognition, neural networks, digital signal processing, and optimization algorithm.

Dr. Papamarkos is a member of the Greek Technical Chamber.

Antonis E. Atsalakis received the Diploma degree in electrical and computer engineering from the Democritus University of Thrace, Xanthi, Greece, in 1999. He is currently pursuing the Ph.D. degree at the Democritus University of Thrace.

Currently, he is a Research and Teaching Assistant at the Department of Electrical and Computer Engineering, Democritus University of Thrace. His research interests include color document image processing and analysis, neural networks, and pattern recognition.

Charalampos P. Strouthopoulos was born in Drama, Greece, in 1962. He received the Diploma degree in electrical engineering from the University of Patras, Greece, and the Ph.D. degree in electrical and computer engineering from the Democritus University of Thrace, Xanthi, Greece, in 1985 and 1999, respectively. His Ph.D. thesis was on document image analysis techniques.

Currently, he is a Research Fellow at the Democritus University of Thrace. His main research interests are in digital image processing, document image analysis, color quantization, and pattern recognition.

Dr. Strouthopoulos is a member of the Technical Chamber of Greece.