Computer Vision and Image Understanding **78,** 336–350 (2000)
doi:10.1006/cviu.2000.0838, available online at http://www.idealibrary.com on IDEAL®

Gray-Level Reduction Using Local Spatial Features

Nikos Papamarkos[1] and Antonios Atsalakis

Electric Circuits Analysis Laboratory, Department of Electrical and Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece

Received January 26, 2000; accepted February 9, 2000

This paper proposes a new method for reduction of the number of gray-levels in an image. The proposed approach achieves gray-level reduction using both the image gray-levels and additional local spatial features. Both gray-level and local feature values feed a self-organized neural network classifier. After training, the neurons of the output competition layer of the SOFM define the gray-level classes. The final image has not only the dominant image gray-levels, but also has a texture approaching the image local characteristics used. To split the initial classes further, the proposed technique can be used in an adaptive mode. To speed up the entire multithresholding algorithm and reduce memory requirements, a fractal scanning subsampling technique is adopted. The method is applicable to any type of gray-level image and can be easily modified to accommodate any type of spatial characteristic. Several experimental and comparative results, exhibiting the performance of the proposed technique, are presented.

Key Words: segmentation; multithresholding; neural networks; PCA; SOFM.

1. INTRODUCTION

Reduction of the number of gray-levels in an image is an important task for segmentation, compression, presentation, and transmission of images. In most cases, it is easier to process and understand an image with a limited number of gray-levels. The usual technique for reduction of the number of gray-levels in a digital image is multithresholding. Using only the values of the image histogram, multithresholding determines appropriate threshold values that define the limits of the image gray-level classes. However, the application of multithresholding is based on the assumption that object and background pixels in a digital image can be well distinguished by their gray-level values [1]. Therefore, in complex images, such as natural, texture, or poorly illuminated images, multithresholding techniques do not give satisfactory results. Also, in multiobject images, there are several difficulties

[1] To whom correspondence should be addressed. Fax: +30-541-26947. E-mail: papamark@voreas.ee.duth.gr.

1077-3142/00 $35.00

for multilevel threshold selection that are associated with gray-level distributions, small objects, and object overlapping.

Multithresholding techniques can be classified into three categories. In the first category belong the histogram-based multithresholding techniques [2–6]. These techniques use different criteria such as minimum entropy, interclass variance between dark and bright regions, changes of zero-crossing, and curve fitting. The method of Reddi *et al*. [2] is fast and is a version extended to multithresholding, of the global threshold method of Otsu [3], which is one of the most powerful methods for global thresholding [7]. The criterion used is the selection of thresholds so that the interclass variance between dark and bright regions is maximized. The method of Kapur *et al*. [4] is based on the maximum entropy criterion. The method of Carlotto [5] determines thresholds by handling the information derived from the changes of zero-crossings in the second derivative. The method gives good results only for the histograms that satisfy the basic hypothesis that the histogram consists only of univariate normal distributions. The method of Papamarkos *et al*. [6] is based on a combination of a hill-clustering algorithm and an appropriate linear programming approximation technique. In the second category belong methods based on edge matching and classification [8, 9]. These methods are applicable to images with good edges. As a first step, in all these methods, the pixels of the initial image are first classified as edge and nonedge pixels by using an edge extraction algorithm. Consequently, for the extraction of the best thresholds, computationally expensive recursion procedures are used. During each iteration, the threshold values are modified to satisfy some edge characteristics. Finally, in the third category belong all the other techniques, which can be characterized as hybrid. Spann and Wilson [10] propose a hybrid method which is a combination of a quad-tree smoothing technique, a local centroid clustering algorithm, and a boundary estimation approach. This method is applicable under the condition that the histogram consists only of Gaussian distributions.

Other approaches to gray-level reduction (GLR) are based on nearest-gray-level merging or on gray-level error diffusion. In the nearest-gray-level methods, each pixel in the image changes its value to the gray-level in a palette that is the closest match to some typical neighboring pixel. The error diffusion techniques are based on dithering approaches. The "error" refers to the cumulative difference between the actual values of pixels in the image and their "true" values, if all were set to their correct gray-levels. These techniques are suitable for elimination of uncommon gray-levels in an image but are ineffective for image analysis and segmentation. It should be noticed that none of the multithresholding techniques takes into account the local texture characteristics of an image.

This paper proposes a new gray-scale reduction algorithm, which exploits not only the gray-levels of the pixels but also the local spatial characteristics of the image. The proposed approach significantly improves a previously reported technique [11, 12], which uses only the gray-level values of the images to perform GLR. According to the proposed technique, the gray-level value of each pixel is related to local spatial features extracted in its neighboring region. Thus, the one-dimensional histogram-clustering approach of the multithresholding techniques is now converted to a multidimensional feature-clustering technique. The gray-level value of each pixel is considered the first feature. 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, and entropy values. The feature set feeds a self-organized neural network classifier, which consists of a PCA and a Kohonen SOFM [13, 14]. The PCA is used to manipulate the feature 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 Kohonen's learning algorithm. After training, the output neurons of the SOFM define the appropriate feature classes. Next, each pixel is classified into one of these classes and resumes the gray-level of the class. In this way, the original image is converted into a new one, which has a limited number of gray-levels and whose spatial characteristics approximate those defined by the features used. To reduce the storage requirement and computation time, the training set can 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 [15].

The use of additional spatial features permits the application of the GLR technique in an adaptive mode. According to this scheme, the GLR technique is initially applied without using spatial features and the gray-levels are classified into m classes. Next, the pixels of the initial image that belong to each one of the m classes are further classified by the GLR algorithm using additional spatial feature sets. It is obvious that by increasing the feature space the gray-level classes can be further split. This procedure is repeated level by level. Figure 5 depicts the proposed technique in a binary tree form.

The proposed method was tested with a variety of images and the results are compared with other multithresholding techniques. In addition, this paper presents characteristic examples for various image local characteristics. The experimental and comparative results confirm the effectiveness of the proposed method.

2. DESCRIPTION OF THE METHOD

A digital gray-scale image $I(i, j), i = 1, \ldots, n, j = 1, \ldots, m$, can be considered a set of $n \times m$ pixels, where each pixel is a point in the gray-scale space. Usually, the total number of gray-levels is restricted to 256; i.e., $I(i, j) \in [0 \ldots 255]$.

Let $N(i, j)$ denote the local region neighboring pixel (i, j). In this approach $N(i, j)$ is considered to be a 3×3 or a 5×5 mask that has as its center the pixel (i, j). It is assumed that every pixel (i, j) belongs to its neighborhood $N(i, j)$. It is obvious that in most cases, the gray-level of each pixel is associated with the gray-levels of the neighboring pixels and the local texture of the image. Therefore, the gray-level of each pixel (i, j) can be associated with local image characteristics extracted from the region $N(i, j)$. These characteristics can be considered local spatial features of the image and can be helpful for the GLR process. That is, using the gray-level values of $N(i, j)$, $f_k, k = 2, 3, \ldots, K + 1$, local features can be defined, which are next considered as image spatial features. As can be observed in Fig. 1, each pixel (i, j) is related to its $I(i, j)$ gray-level value, which is considered the first spatial feature, and to K additional features. This approach transforms the 1-D gray-scale feature space of a standard multithresholding technique to a more advantageous one of $K + 1$ dimensions. No restrictions on the type of local features are implied. However, the features must represent simple spatial characteristics, such as min, max, entropy, and median values of the neighboring masks.

According to the above analysis, the gray-scale reduction problem can be considered the problem of best transforming the original gray-level image to a new one with only J gray-levels as the final image, to approximate not only the principal gray-level values but also the local characteristics used. An effective approach to this problem is to consider it a clustering problem and achieve its solution using a suitable self-organized neural network.

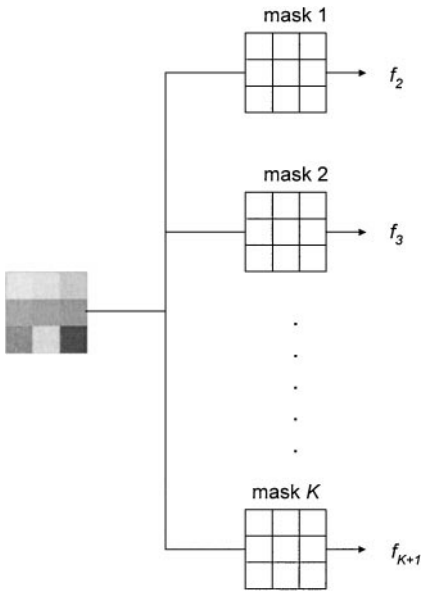


FIG. 1. The feature extraction procedure.

In the proposed approach, the topology of the entire neural network has the structure shown in Fig. 2. As can be observed, it consists of a PCA and a SOFM neural network. PCA is useful because of the multidimensionality of the feature space. Through the PCA transform, 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 a SOFM neural network 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 succeed. In other words, a SOFM neural network decreases the input feature space optimally into a smaller one. The

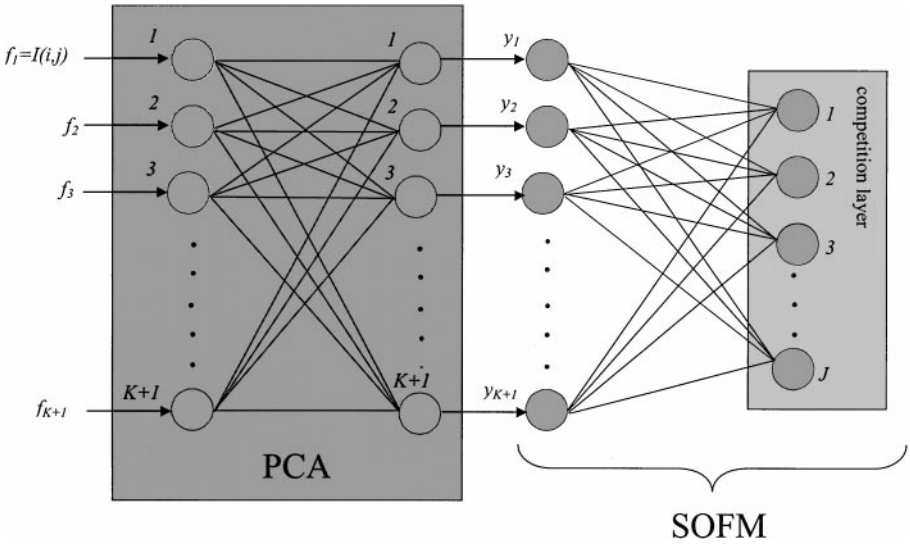


FIG. 2. The entire neural network.

resultant feature space can be viewed as a representative of the original feature space and therefore it approximates the main statistical characteristics of the input space.

The PCA has $K+1$ input and $K+1$ output neurons. It is used to increase the discrimination of the feature space. The SOFM has $K+1$ input and J output neurons. The entire neural network 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 neural network, each image pixel is classified into one of the J classes and converts its gray-level to the gray-level defined by the class. To speed up the algorithm and minimize its memory requirements, a fractal scanning subsampling procedure can be used.

2.1. The PCA Neural Network

As mentioned above, the first stage of the neural network is a PCA. After training, the feature vector is optimally projected to a new set such that the maximum discrimination in the feature space is obtained. For each pixel, the gray-level and its local features contribute proportionally to the training set. 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 the intrinsic information contained in the data. In multivariate data analysis, the Karhumen–Loeve transformation (KLT) [14] is widely used for PCA. It is a procedure for the estimation of the eigenvalues and eigenvectors requiring the computation of a data covariance matrix. Here, a single-layer feedforward neural network is used to perform a PCA. Using this technique, there is no need to compute the covariance matrix, since the eigenvectors are derived directly from the data. In our approach, the input vector is the $(K+1)$-dimensional $\boldsymbol{I}$, and the output is also a $(K+1)$-dimensional feature vector $\boldsymbol{y}$, which is computed by the relation

$$\boldsymbol{y} = \boldsymbol{Q}\boldsymbol{I}. \tag{1}$$

The input of the PCA (input features) is the gray-level value of each pixel of the image and the values of the local features. The output consists of $K+1$ neurons. Matrix $\boldsymbol{Q}$ provides the PCA neural network coefficients. The net is trained by the GHA. 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, then the strength of that synapse is selectively increased; otherwise it is weakened or eliminated. The GHA is implemented in the training stage via the relation

$$\Delta q_{ik} = \gamma\left(\underbrace{y_i I_k}_{(A)} - \underbrace{y_i \sum_{n=0}^{i} q_{nk} y_n}_{(B)}\right), \quad \text{with } i, k = 1, 2, 3, \ldots. \tag{2}$$

Term (A) expresses the Hebbian rule and term (B) imposes a limit on the growth of synaptic weights. It should be noticed that after training $\boldsymbol{Q}$ approximates with probability one the matrix whose rows are the three eigenvectors of the covariance matrix, formed by decreasing eigenvalues. Also, for example, the weight coefficients q_{00}, q_{01}, and q_{02} of the first neuron determine an eigenvector which exhibits the maximum discrimination power in its direction. The output values y_0, y_1, and y_2 of the PCA are the projections of the input vector onto these eigenvectors. Thus, the training of SOFM is based on the projections of feature vectors.

2.2. The SOFM Neural Network

The structure of the Kohonen SOFM neural network used is depicted in Fig. 2. It has $K+1$ input and J output neurons, arranged in a one-dimensional 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 gray-level values but also with the spatial features used. Adjusting the number of the output neurons, we can define the number of gray-levels in the final image. The output J neurons are related to input neurons via the $w_{i,j}$, $i=1,\ldots,K+1$ and $j=1,\ldots,J$ coefficients.

The SOFM is competitively trained according to the weight update function

$$\Delta w_{ji} = \begin{cases} a(y_i - w_{ji}), & \text{if } |c-j| \le d \\ 0, & \text{otherwise} \end{cases}, \quad i=1,\ldots,K+1 \text{ and } j=1,\ldots,J, \tag{3}$$

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

After training, the optimal gray-level values obtained by the neural network are equal to

$$w_{i,j}, \quad j=1,\ldots,J. \tag{4}$$

Next, the original image is rescanned and by using the neural network, the new image is constructed with only J gray-level values in a way that approximates the texture of the original image, according to the spatial features used. However, if we are not interested in the spatial characteristics, then we can feed and train the neural network using only the gray-level values. Obviously, this procedure results in a reduced set of gray-levels, which are close (according to Kohonen's learning rule) to the gray-level distribution of the original image.

2.3. Image Subsampling

The proposed technique can be applied to gray-level images without any subsampling. However, in the case of large images and in order to achieve reductions in computational time and memory size requirements, it is preferable to have a subsampled version of the original image. For the subsampled image to be a better representation (to capture the image texture better) than the original image, we choose to use a fractal scanning process, based on the well-known Hilbert space-filling curve. To generate the Hilbert curve we must start with the basic staple-like shape as shown in Fig. 3a. The rest of the Hilbert curve is created

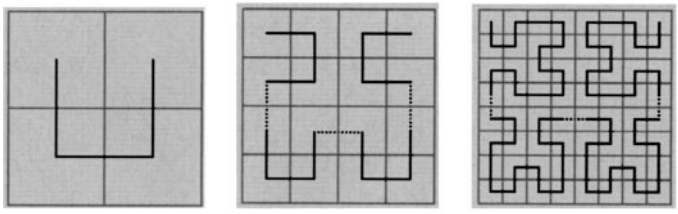


FIG. 3. Generation of the Hilbert curve.

FIG. 4. Image subsampling using fractal scanning.

sequentially, using the same algorithm. Starting with the basic Hilbert curve, we increase the grid size in each step by a factor of 2. 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 90° counterclockwise and the upper right quarter is rotated 90° clockwise. Finally, we connect the four pieces with short straight segments to obtain the next step curve. This process is depicted in Figs. 3b and 3c. A complete description of the Hilbert curve is given in [15]. If k is the order of the Hilbert curve, then the ratio of the subsampled image pixels to the total number of pixels is given by

$$\frac{4^k}{\text{number of image pixels}}. \tag{5}$$

As an example, Fig. 4 depicts a fractal image subsampling of a 200×200-pixel image with $k = 7$.

2.4. The Stages of the Method

In summary, according to the above analysis, the stages of the proposed GLR technique are:

Stage 1. Define or determine the number J of final gray-levels and the number and type of spatial features.

Stage 2. Construct the PCA and SOFM neural networks.

Stage 3. Define the subsampling parameters.

Stage 4. Train the entire neural network.

Stage 5. Using the neural network, transform the original image to a new one with the reduced number of gray-levels obtained in Stage 4.

3. ADAPTIVE SCHEME

In some cases, where the gray-level classes are not properly separated, the GLR technique can be applied in an adaptive mode. This procedure follows a tree structure (Fig. 5) with

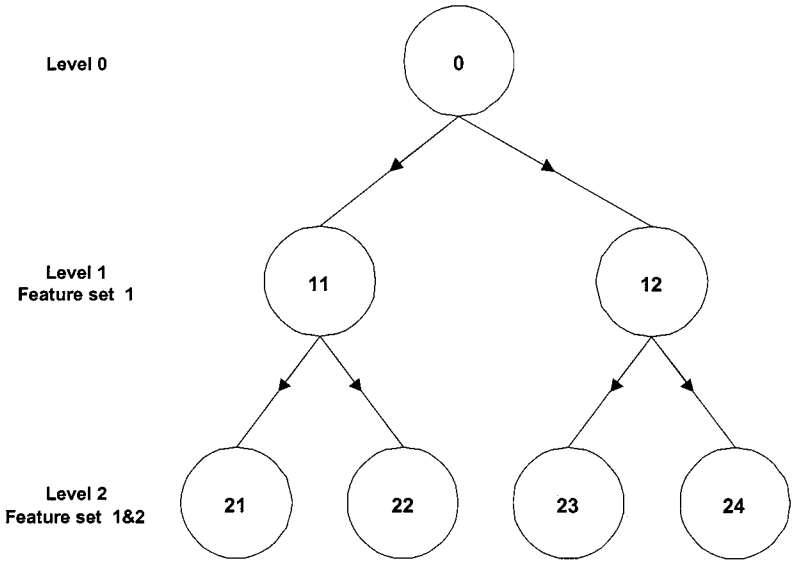


FIG. 5. The tree scheme for the adaptive application of the GLR technique.

levels and nodes. At each level, an additional and appropriate set of features can be used so that new classes become visible. In each tree node, the GLR technique is performed only on the pixels of the initial image that correspond to the gray-level class produced at the previous level. The entire procedure is terminated when we come to a predefined level of the tree. In the final stage, the extracted images are merged by a simple AND procedure.

4. EXPERIMENTAL RESULTS

The proposed method has been implemented using Borland Delphi 4. The PCA and the SOFM neural networks are implemented and tested using the developer's version 3.02 of the neural network package NeuralSolution (NeuroDimension, Inc.).

Experiment 1. Let us consider the original 256 gray-level image shown in Fig. 6a. The image size is 512×512 pixels and its resolution is 72 dpi. Its histogram is depicted in Fig. 6b. In this example, the proposed method uses two spatial features described by the following mask operations:

$$f_2(i, j) = 10 \cdot I(i, j) - \sum_{\substack{a=-1 b=-1 \\ (a,b)\neq(0,0)}}^{1}\sum^{1} I(i + a, j + b) \tag{6}$$

$$f_3(i, j) = 8 \cdot I(i, j) - \sum_{\substack{a=-1 b=-1 \\ (a,b)\neq(0,0)}}^{1}\sum^{1} I(i + a, j + b), \quad \text{with bias} = 255. \tag{7}$$

The feature f_2 is an edge extraction mask, whereas feature f_3 can be characterized as a contour extraction mask. It should be noticed that the two features used can be considered to be of similar type because they enhance the image edges.

The proposed GLR method is applied with only four neurons in the SOFM competition layer. Therefore, the resultant image will have only four gray-levels. The subsampling set

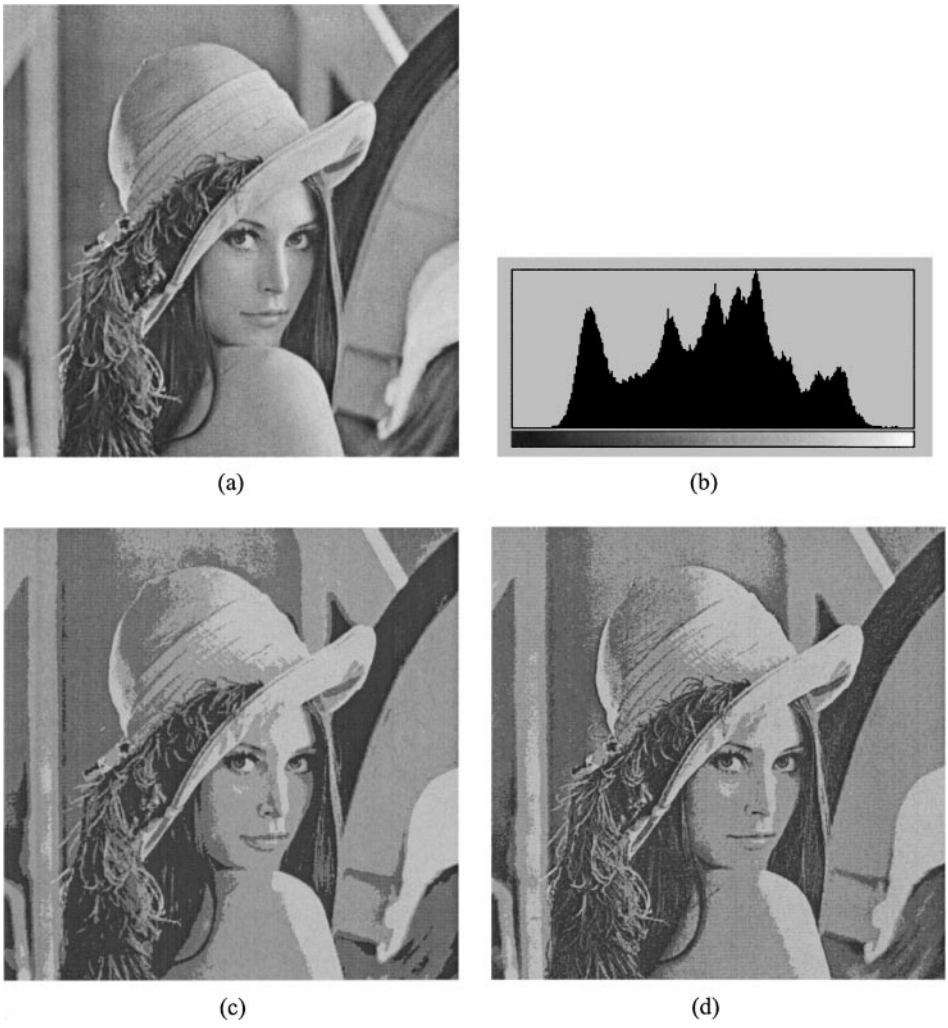


FIG. 6. (a) The original Lenna image. (b) The image histogram. (c) The image with four gray-levels without use of spatial features. (d) The image with four gray-levels obtained using spatial features.

of pixels, according to Eq. (5) for $k = 7$, consists of 16,384 pixels, instead of the 262,144 pixels in the original image. For comparison, the proposed algorithm is first applied without using spatial features. In this case, the four gray-level values obtained are 55, 107, 149, and 196 and the converted image is shown in Fig. 6c. In contrast, using the spatial features, the proposed GLR method leads to gray-levels 54, 96, 143, and 193. Figure 6d shows the final image obtained. Comparing Fig. 6c to Fig. 6d, we observe that Fig. 6c has flat and more uniform areas and that the local characteristics of these images are significantly different.

In order to have some comparative GLR results, we apply to the same image the multi-thresholding algorithms of Reddy *et al*., Kapur *et al*., and Papamarkos *et al*. and the decrease a color utility of PaintShop Pro 5. Table 1 summarizes the results obtained with these techniques and Fig. 7 shows the different images obtained. It can be observed that in any case the proposed method leads to an image which better approximates the spatial characteristics used. This is obvious, for example, in the lower left region of the image.

TABLE 1
Comparative Results

Method	Final gray-level values
Proposed	54, 96, 143, 193
Proposed without spatial features	55, 107, 149, 196
Reddi *et al*.	40, 102, 147, 212
Kapur *et al*.	12, 60, 130, 209
Papamarkos *et al*.	36, 92, 150, 222
PaintShop Pro 5	53, 111, 152, 210

Experiment 2. This second example demonstrates the influence of different types of spatial features on the 584×386-pixel image of Fig. 8a. Figure 8b shows the results obtained for five gray-levels, without use of spatial features. In this example, four different types of spatial features are applied independently. The first feature is the contour

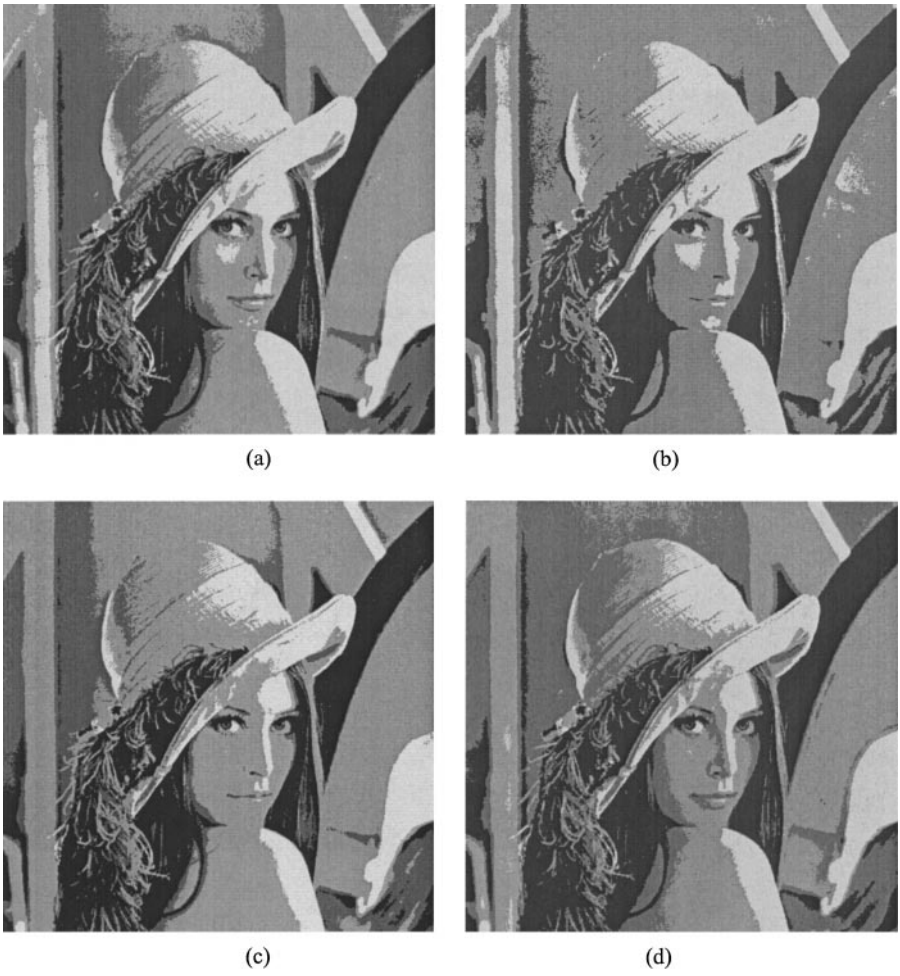
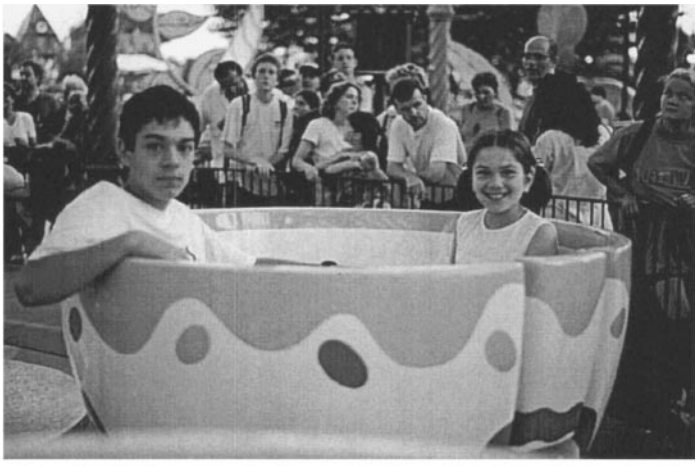
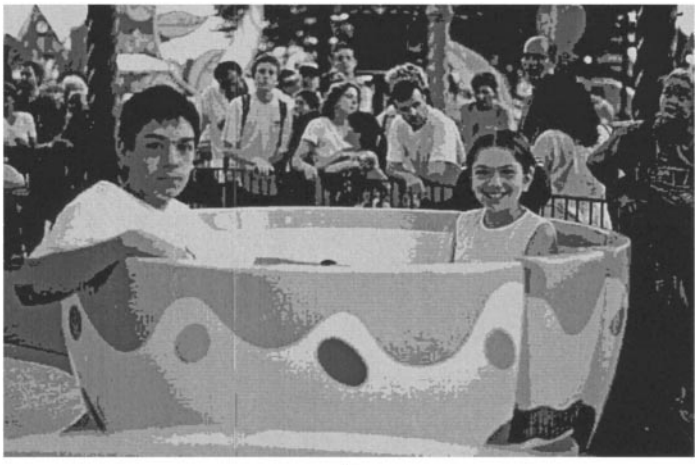


FIG. 7. Images obtained using the method of (a) Reddi *et al.*, (b) Kapur *et al.*, (c) Papamarkos *et al.*, and (d) PaintShop Pro 5.



(a)



(b)

FIG. 8. (a) The original image. (b) The image with five gray-levels without using spatial features.

feature described by Eq. (7). The second feature is extracted using the Kirsch mask [16]:

$$f_2(i,j) = 3[I(i-1,j-1)+I(i,j-1)+I(i+1,j-1)+I(i+1,j)+I(i+1,j+1)] - 5[I(i-1,j)+I(i-1,j-1)+I(i,j+1)]. \quad (9)$$

The Kirsch operators approximate the image first derivative.

The last two features are extracted by using the nonlinear operations min and max:

$$f_2(i,j) = \text{minimum}\begin{bmatrix} I(i-1,j-1) & I(i,j-1) & I(i+1,j-1) \\ I(i-1,j) & I(i,j) & I(i+1,j) \\ I(i-1,j+1) & I(i,j+1) & I(i+1,j+1) \end{bmatrix} \quad (10)$$

$$f_2(i,j) = \text{maximum}\begin{bmatrix} I(i-1,j-1) & I(i,j-1) & I(i+1,j-1) \\ I(i-1,j) & I(i,j) & I(i+1,j) \\ I(i-1,j+1) & I(i,j+1) & I(i+1,j+1) \end{bmatrix}. \quad (11)$$

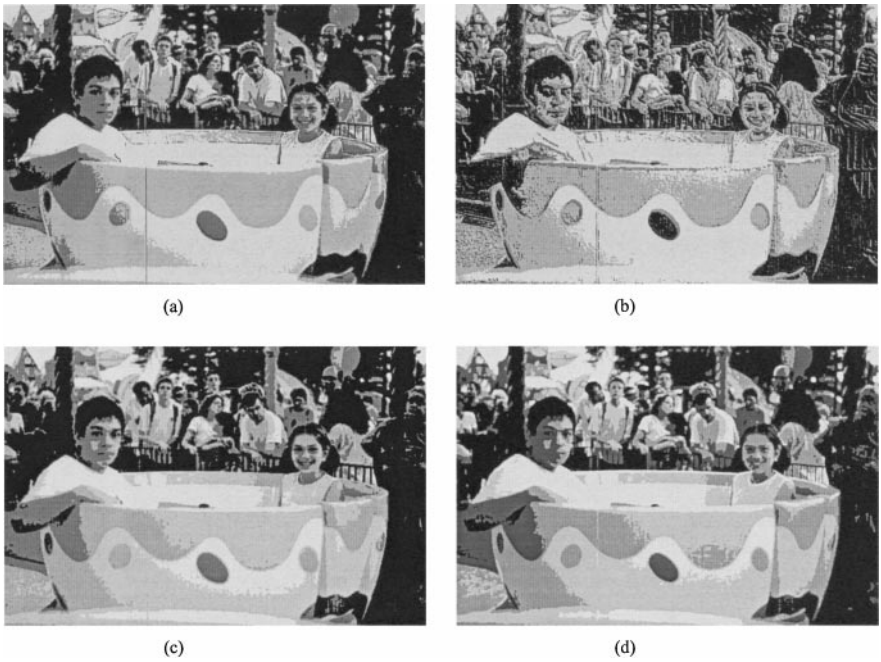


FIG. 9. Images obtained using (a) contour, (b) Kirsch, (c) min, and (d) max spatial features.

The min operator spreads out black areas and shrinks white areas. The max operator spreads out white areas and chokes black areas. Figure 9 shows the final images obtained in the four cases. In order to clarify the influence of the feature type, Fig. 10 shows the images obtained by the application of the contour, Kirsch, min, and max operators. It can be easily

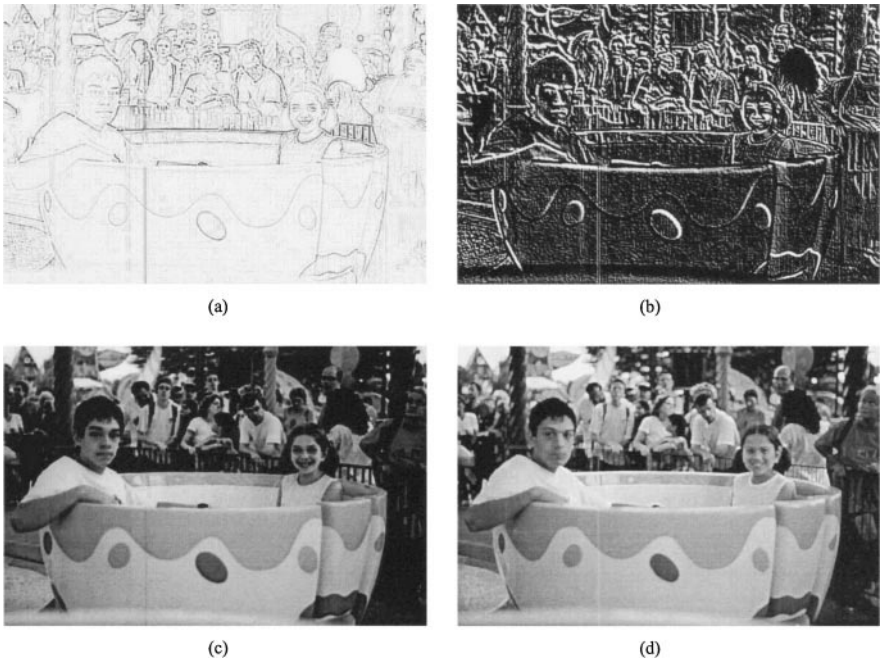


FIG. 10. Images obtained by application of (a) contour, (b) Kirsch, (c) min, and (d) max operators.

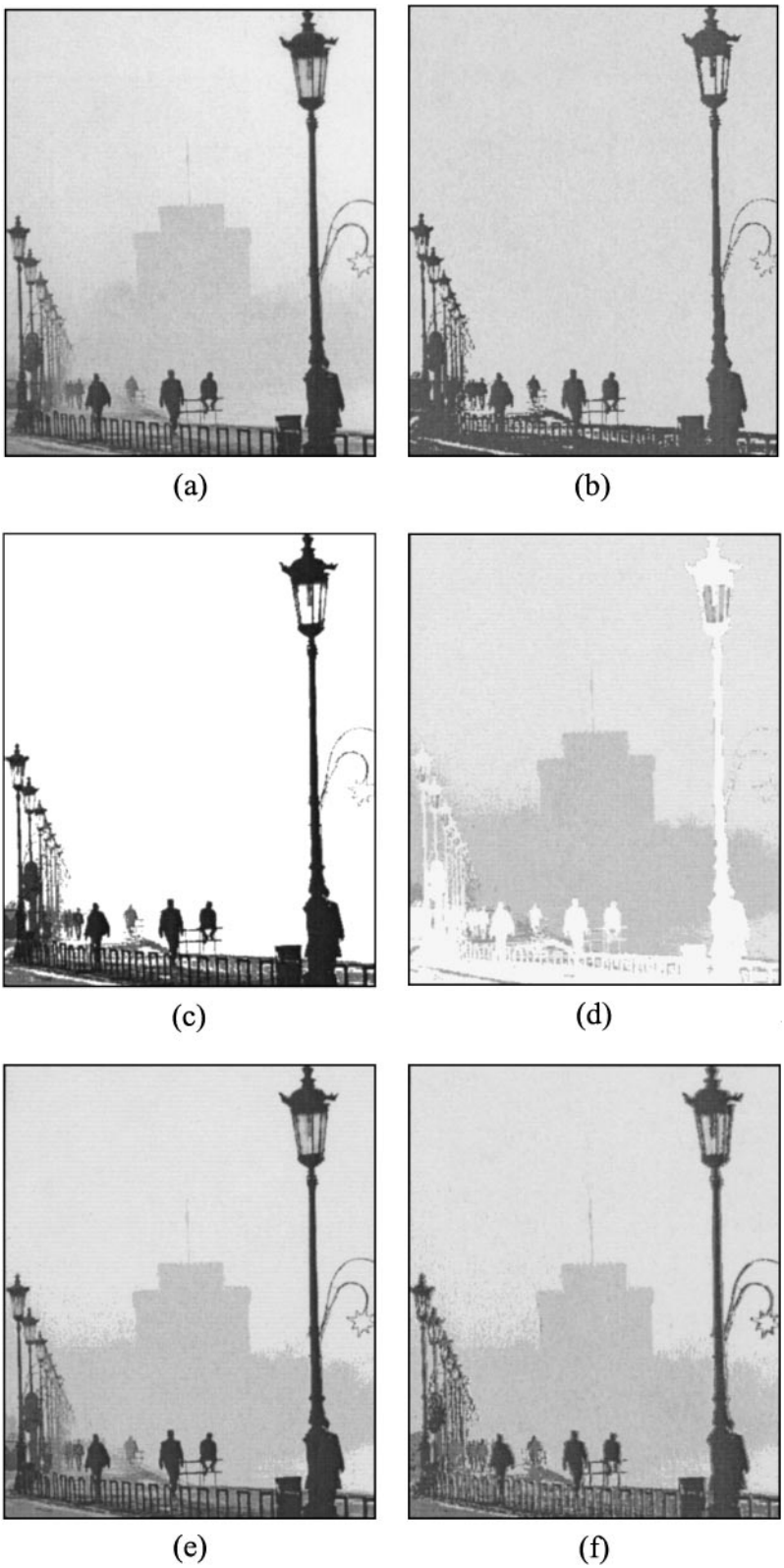


FIG. 11. (a) Original image, (b) bilevel image, (c) first image of level 1, (d) second image of level 1, (e) final image, and (f) image obtained without using the adaptive process.

observed that the texture of the final images obtained approaches the texture of the images of Fig. 10.

Experiment 3. This experiment demonstrates the application of the adaptive mode of the GLR algorithm to the test image shown in Fig. 11a. The GLR algorithm will be applied using only one level of a tree. In the root node (level 0), after 7 s, the application of the GLR technique results in dominant gray-level values of 97 and 236 and the original image is converted to the image of Fig. 11b. At the next level, the pixels of the initial image that correspond to the dominant gray-level value of 97 in Fig. 11b are used to construct a new image. In this image, the GLR algorithm is applied using as additional features the emboss features derived by

$$f_2(i, j) = -[I(i-1, j-1) + I(i, j-1) + I(i+1, j-1)] + I(i, j) + I(i+1, j+1) + I(i, j+1) + I(i-1, j+1) \quad (9)$$

and for only three gray-level classes. This leads to the image shown in Fig. 11c, which has only the dominant gray-level values of 69, 101, and 143. Similarly, on the right branch of the tree, the image obtained and shown in Fig. 11d has only the gray-level values of 184, 220, and 244. Merging the images of Figs. 11c and 11d we obtain the final image of Fig. 11e. For comparison reasons, Fig. 11f shows the image obtained by the application of the GLR algorithm in its simple form (without adaptation) for six gray-level classes using the emboss features. The gray-level values are calculated to be equal to 85, 124, 131, 159, 219, and 243. Comparing Figs. 11e and 11f, we observe that the adaptive form of the GLR algorithm gives better gray-level reduction results.

It should be noticed that in all cases of this experiment, the number of subsampled pixels via Hilbert space-filling curve was equal to 20.000.

5. CONCLUSIONS

This paper proposes a general GLR technique, which is applicable to any gray-level image. The proposed technique is based on a neural network structure which consists of a PCA and a Kohonen SOFM. The training set of the neural network consists of the image gray-level values and additional spatial features extracted in the neighborhood of each pixel. These features describe image local characteristics. Therefore, the gray-level of each pixel is related to the gray-levels and the texture of the neighboring pixels. Therefore, the final image has the proper gray-levels, and its structure approximates the local characteristics used. Additionally, the proposed GLR algorithm can be applied in an adaptive mode. In this case, different feature sets can be applied at every level of the adaptation procedure. In order to speed up the entire algorithm, a fractal subsampling, based on the Hilbert space-filling curve, can be used.

The proposed method was successfully tested with a variety of images. The experimental results presented in this paper show the efficiency of the method.

REFERENCES

1. J. Kittler and J. Illingworth, Minimum error thresholding, *Pattern Recognition* **19**, 1986, 41–47.
2. S. S. Reddi, S. F. Rudin, and H. R. Keshavan, An optimal multiple threshold scheme for image segmentation, *IEEE Trans. Systems Man Cybernet.* **14**(4), 1984, 661–665.

3. N. Otsu, A threshold selection method from gray-level histograms, *IEEE Trans. Systems Man Cybernet*. **9**(1), 1979, 62–69.
4. J. N. Kapur, P. K. Sahoo, and A. K. Wong, A new method for gray-level picture thresholding using the entropy of the histogram, *Comput. Vision Graphics Image Process*. **29**, 1985, 273–285.
5. M. J. Carlotto, Histogram analysis using scale-space approach, *IEEE Trans. Pattern Anal. Mach. Intell*. **9**(1), 1987, 121–129.
6. N. Papamarkos and B. Gatos, A new approach for multithreshold selection, *CVGIP: Graph. Models Image Process*. **56**, 1994, 357–370.
7. P. K. Sahoo, S. Soltani, and A. K. C. Wong, A survey of thresholding techniques, *Comput. Vision Graphics Image Process*. **41**, 1988, 233–260.
8. S. Wang and R. M. Haralick, Automatic multithreshold selection, *Comput. Vision Graphics Image Process*. **25**, 1984, 46–67.
9. L. Hertz and R. W. Schafer, Multilevel thresholding using edge matching, *Comput. Vision Graphics Image Process*. **44**, 1988, 279–295.
10. M. Spann and R. Wilson, A quad-tree approach to image segmentation which combines statistical and spatial information, *Pattern Recognition* **18**, 1985, 257–269.
11. C. Strouthopoulos and N. Papamarkos, Text identification for image analysis using a neural network, in *Special Issue on Image Processing and Multimedia Environments Image Vision Comput*. **16**, 1998, 879–896.
12. N. Papamarkos, C. Strouthopoulos, and I. Andreadis, Multithresholding of color and gray-level images through a neural network technique, *Image Vision Comput.* **18** (2000), 213–222.
13. T. Kohonen, *Self-Organizing Maps*, Springer-Verlag, New York, 1997.
14. S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan Co., New York, 1994.
15. H. Sagan, *Space-Filling Curves*, Springer-Verlag, New York, 1994.
16. M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision, 2nd ed.*, Brooks/Cole, Pacific Grove, CA, 1999.