Neural-based color image segmentation and classification using self-organizing maps

JANDER MOREIRA [1]
LUCIANO DA FONTOURA COSTA [2]

[1] Universidade Federal de São Carlos
Departamento de Computação
GAPIS - Architecture, and Image & Signal Processing Research Group
jander@dc.ufscar.br

[2] Universidade de São Paulo
IFSC - Instituto de Física de São Carlos
Cybernetic Vision Research Group
Luciano@olive.ifqsc.sc.usp.br

Abstract. This paper presents a method for color image segmentation which uses classification to group pixels into regions. The chromaticity is used as data source for the method because it is normalized and considers only hue and saturation, excluding the luminance component. The classification is carried out by means of a self-organizing map (SOM), which is employed to obtain the main chromaticities present in the image. Then, each pixel is classified according to the identified classes. The number of classes is *a priori* unknown and the artificial neural network that implements the SOM is used to determine the main classes. The detection of the classes in the SOM is done by using a K-means segmentation. The obtained results substantiate the feasibility of the method, whose performance is compared, for evaluation, to human-assisted segmentation. A comparison of the method with a segmentation based on the *k*-nearest-neighbor classification is also presented.

Keywords: color segmentation, neural networks, self-organizing maps, classification, K-means segmentation, nearest-neighbor classification.

Introduction

One of the most important stages in artificial vision systems is segmentation. Since vision is difficult to be achieved, biological inspiration is an interesting approach toward the development of more versatile, cybernetic, vision systems. A system as such, named *Cyvis-1* [Costa et al. (1994)] (that stands for *Cybernetic Vision System*, first version), consists of a collection of processing specialization or modules, each performing dedicated tasks and sharing information with other modules. The main tasks incorporated include depth examination (through stereo analysis), edge and region detection, texture, and color labeling. These cooperative modules generate an intermediate description of the scene to be processed at a higher level, and also produce control information to correct and adjust camera operation (active vision).

Image segmentation is the process of division of the image into regions with similar attributes [Pratt (1991)]. This endeavor uses many techniques developed during the past decades, most of them dealing with binary and monochromatic images. The use of color information in image processing is only nowadays emerging, with the availability of cheaper technology. The whole information present in color images is not available in grayscale versions. Colors are one of the most important features considered in biological visual systems, since it is used to discern objects and patterns, even in conditions of equiluminance [Levine (1985)]. Since humans are trichromats, which means that most of the visible color can be produced by a composition of three independent light colors, digital color images usually have three components: *red*, *green* and *blue*.

The approaches for the segmentation of digital color images can be broadly classified in: (a) edge and line oriented segmentation, (b) region growing methods, (c) clustering, and (d) region splitting methods [Ohta et al. (1980)]. Edge and line oriented segmentation work on image data either individually analyzing each data band (e.g., the RGB channels) or considering the whole vector space, by using gradient calculated in a two-dimensional vector space [Cumani (1991)] [Lee & Cok (1991)]. After edge extraction, a postprocessing should be applied to create the objects and segments that represent the elements present in the scene.

Region growing and splitting methods deals commonly with feature extraction and thresholding. The approach by Ohta, Ohlander and Sakai [Ohta et al. (1980)] uses the Karhunen-Loève transformation to extract a significant color feature set, adopted in a region splitting phase. A similar approach is stated by Ohlander, Price, and Reddy [Ohlander et al. (1978)] that uses significant threshold to split regions. The histogram is calculated for each color component and the best peak is detected and adopted to split the regions.

Clustering techniques for color images often deals with fuzzy approaches. A largely employed method is clustering using the fuzzy *c*-means to detect the different classes [Huntsberger et al. (1985)] [Huntsberger et al. (1986)] [Lim & Lee (1990)]. The restriction is that the number of classes, in most cases, must be know *a priori*. Another similar approach make use of the fuzzy *c*-means clustering in distinct resolution levels [Trivedi & Bezdec (1986)]. Also, the computational cost can be reduced with a coarse-to-fine technique, in which the histograms are analyzed by a scale-space based tool. Another work addresses non-fuzzy clustering, i.e., classical techniques, like circular-cylindrical decision elements to cluster features in the 1976 CIE (L^*, a^*, b^*)-uniform color coordinate system [Celenk (1990)].

A segmentation based on classification was proposed by Moreira and Costa, which makes use of a classifier that implements the *k*-nearest-neighbor technique. This method, which requires supervision, considers sample points provided by the operator in order to identify the classes. [Moreira & Costa (1995)]

Artificial neural networks are increasingly being employed in various fields, including signal processing, pattern recognition, medicine, speech production and recognition, and business. They constitute an information-processing system that share conceptual structures and characteristics with biological neural networks [Anderson (1995)][Hertz et al. (1991)]. A recent work that makes use of a multi-layer neural network shows an application in the classification of multispectral remote sensing data [Schlünzen et al. (1994)]. Also an approach to compress color image data using a SOM is proposed, with the intend to generate a “codebook” for color quantization for data transmission [Godfrey & Attikiouzel (1992)].

The present work, which is part of the *Cyvis-1* approach, deals with the color subsystem and describes an attempt to segment color images based on an self-organizing neural network, which is able to recognize the main components (chromaticities) present in images. The next step consists in identifying these components and classifying the image data based on the detected classes. The next session describes the topologic self-organizing maps. Then the developed segmentation method is explained, followed by discussion and experimental results.

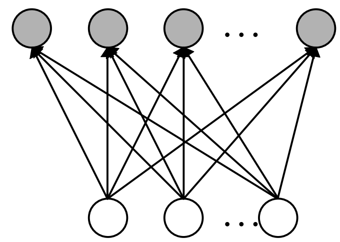


Figure 1. Self-organizing map.

Self-organizing maps

Artificial neural networks are processing systems analogous to biological neural networks, presenting neurons, axons, dendrites, neural layers, transfer functions, and so on. Their paradigms fall in three main categories: supervised, reinforced and self-organized. This classification takes into account the amount of data needed for the training phase. Supervised networks use a previous knowledge about the desired outputs, in such a way that the error between the actual input and expected output is a suitable parameter. Reinforced networks rely on the measure of the overall error but do not need their exact output. Self-organizing networks determine by themselves the internal weight representation for the presented input data and do not need supervision.

The self-organizing neural networks, also known as Kohonen networks [Kohonen (1989)], are networks which incorporate a topology scheme, i.e., take into account the a topological structure among units. The input signals are *n*-tuples and there is a set of *m* cluster units. Each input is fully connected to all units, which respond differently to the input pattern. At each step in the training phase, the cluster unit with weights that best match the input pattern is elected the winner (usually in a minimum Euclidean distance sense). This winning unit and a neighborhood around it are then updated in such a way that their internal weights be closer to the presented input. The adopted updating factor is not equal for all neurons, but stronger near the winning unit, decreasing for more distant units. Figure 1 shows the basic structure of self-organizing maps. There are input components (white circles) connected to all cluster units (shaded circles). The cluster units can assume any spatial distribution, which are usually linear or planar arrays. Weights are associated to each connection. With time, the gain factor must be reduced and also the neighborhood decreases in size.

During the learning phase the node weights are changed in an ordered manner, in such a way that the main image features tend to be organized according to a topological distribution in the network. Adjacent nodes respond similarly, while distant nodes respond diversely. The convergence of the features in the self-organizing map occurs considering some limitations on the gain factor while updating the weights [Yin & Allinson (1995)].

Classification method

The proposed segmentation utilizes a self-organizing map to detect the main features present in the image. The features are represented by their chromaticity values, which express colors hue and saturation, avoiding the luminance component. The chromaticity is obtained by normalizing the RGB components of the image, named the *r*, *g* and, *b* components. Equation (1) shows how the chromaticity components can be obtained from the RGB components.

$$\begin{cases} r = \dfrac{R}{R+G+B} \\ g = \dfrac{G}{R+G+B} \\ b = \dfrac{B}{R+G+B} \end{cases} \qquad (1)$$

The network is composed of an orthogonal grid of cluster units, each associated to three internal weights for the chromaticity data. The initial values for weights are set to small random values before the learning phase. When an input sample is presented to the network, a search is made to choose the winning unit *c*, as stated in (2). The input vector **x** at time *t* is compared to each of the weight vectors $\mathbf{m}_i \in R^n$ and the minimum Euclidean distance between the input signal and the neuron weights determines the closest match.

$$\|\mathbf{x}(t) - \mathbf{m}_c(t)\| = \min_i \{\|\mathbf{x}(t) - \mathbf{m}_i(t)\|\} \qquad (2)$$

Then, the weights are updated considering a circular neighborhood around the winning unit, as showed in (3). A scalar gain function $\alpha = \alpha(t)$ is employed to establish how the updating will be done inside the neighborhood N_c. Outside this neighborhood the weights remain unchanged. The gain function values are $0 < \alpha < 1$ and decrease with time.

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t)[\mathbf{x}_i(t) - \mathbf{m}_i(t)] \\ \forall i \in N_c(t) \qquad (3)$$

The adopted gain function is linear, with the highest values at the winning unit and the lowest values at the borders of the circular neighborhood (Figure 2).

Both the neighborhood radius and gain function decrease in magnitude with time. This is accomplished by two reduction factors, which are multiplied to the radius and the gain function parameters in order to reduce their values. Both values range from 0 to 100%, with zero factor meaning no reduction and 100% meaning immediate shrink. Normally, a small value is required in order to achieve SOM's feature space convergence.

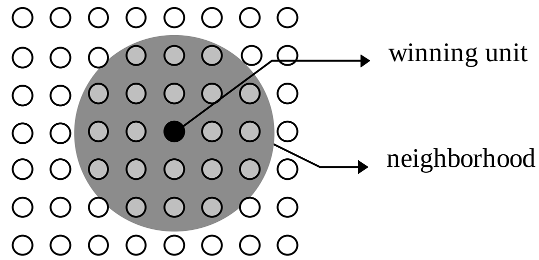


Figure 2. Neighborhood around a winning unit

The feature map final state depends on the following three main conditions: the initial values of the weights in the network, the data set used for training, and the characteristics (parameters) of the map. The initial weights are random and small, and theirs contribution for the final state decreases with the number of samples [Yin & Allinson (1995)]. The sample data is randomly chosen from the image pixels and, when many samples are submitted for network learning, they represent mostly the main chromaticities in the image. Finally, the map configuration parameters, such as the number of nodes of network, how much neighborhood decreases in size with time, and the way weights are updated (the gain function), are the predominant elements determining the final results. These parameters have been evaluated experimentally, and the results are considered in the next session.

The self-organizing map is a topological organizer, in the sense of data ordering, and not a clustering procedure by itself [Pal et al. (1993)]. The next step for segmentation is the analysis of the map to localize the main chromaticities (classes). This is attained considering the final weights vectors in the map as the new sample space. This new data set is used for clustering, allowing the determination of a set of cluster centers.

The clustering process to group the SOM data is based on the method proposed by Coleman and Andrews for image segmentation [Coleman & Andrews (1979)]. With the availability of the features, grouping is performed in order to define the optimum number of clusters along with the center for each cluster. Then, in a subsequent phase, the assignment of each sample data to its closest cluster center is done.

The clustering algorithm determines two initial cluster centers, used to perform a first clustering of the data, in a closest center sense. After this initialization, at each step,

Figure 3. True color RGB image.

a new cluster center is added and a quality measure is calculated. When the quality measure reaches its maximum, the corresponding number of clusters is determined. When the number of clusters is incremented by one, the new cluster center becomes the feature vector that is the further from its closest cluster center.

The clustering quality factor is given by (4), where tr[·] is the trace of a matrix and $\mathbf{S}_W$ and $\mathbf{S}_B$ denotes, respectively, the within-cluster and between-cluster scatter matrices.

$$\beta = \mathrm{tr}\left[\mathrm{S_W}\right]\mathrm{tr}\left[\mathrm{S}_B\right] \tag{4}$$

The scatter matrices are given by:

$$\mathbf{S}_W = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{M_K}\sum_{\mathbf{x}_i \in S_k}\left[\mathbf{x}_i - \mathbf{u}_i\right]\left[\mathbf{x}_i - \mathbf{u}_i\right]^T \tag{5}$$

$$\mathbf{S}_B = \frac{1}{K}\sum_{k=1}^{K}\left[\mathbf{u}_k - \mathbf{u}_0\right]\left[\mathbf{u}_k - \mathbf{u}_0\right]^T \qquad \mathbf{u}_0 = \frac{1}{M}\sum_{i=1}^{M}\mathbf{x}_i \tag{6}$$

In the above equations, K is the number of clusters, M_K is the number of vector elements in the k^{th} cluster, $\mathbf{x}_i$ is a vector element in the k^{th} cluster, $\mathbf{u}_i$ is the mean of the k^{th} cluster, S_k is the set of elements in the k^{th} cluster and $\mathbf{u}_0$ is the mean of all of the feature vectors. M denotes the total number of points to be clustered.

When the cluster centers become available, the original image data is classified to the cluster with the closest center.

Other clustering techniques could also be used for the clustering phase, but they often require previous knowledge of the number of clusters in order to group the samples. When the number of objects is previously known, such techniques can be used. [Duda & Hart (1973)]

Results and discussion

The experiments for segmentation were obtained by the application of the method to the color image in Figure 3. This is an indoor laboratory scene, with fluorescent light illumination. No care was taken to minimize either reflection or other light effects. The image contains four main objects: the red folder, the light-bluish table (on which the other objects relay on), the yellow cow model, and the dark-green book at the upper-right corner of the image. Each object is characterized by its predominant color, which is distinct from each other.

Some details in the image must be also elucidated in order to correctly interpret the obtained results. The cow model in the image contain many colored detail, such as the blue ribbons and its head, which are not yellow. The folder includes an induced texture (attributed to the capture process), causing the color to vary from top to bottom and mainly from left to right; the main color is maintained, but noticeable tonality variations are present. Finally, probably during capture, a high-frequency texture was added to the book (not easily visualized). Due to these consideration, some classification errors are expected.

The chromaticity of the image was calculated by using (1). Random points in the image were then randomly selected to be submitted to the network for the learning phase. Each single sample updates the weights of the nodes in a defined neighborhood as described. After training, the SOM will have a topological distribution of the principal chromaticities in the image. This generated feature map depends on the initial network weight settings, on the random set of chromaticity samples, and on the parameters used in the network.

The initial configuration in the experiments has not been considered a determinant factor for the results. The initial weights tended only to produce, in general, symmetric or rotated versions of an equivalent map, with the relation among final values neighbors units being maintained. The visual inspection of the produced maps, represented by the “colors” of the weights, lead to this conclusion. The term “colors” was used in the sense that each cluster unit has a 3-D vector for weight representation, with each vector component related to a chromaticity component (r, g, and b, respectively); so such information was used to visualize the network array on a color display.

The number of samples used to generate a feature map can range from approximately 800 to 15000, with coherent results. Fewer samples lead to uncomplete maps,

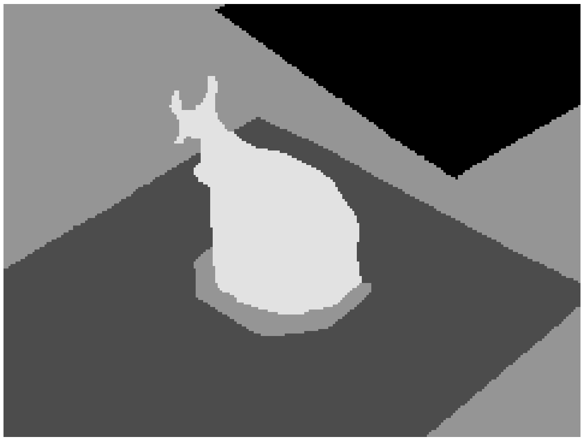


Figure 4. Human-assisted reference segmented image used for comparison with obtained results.

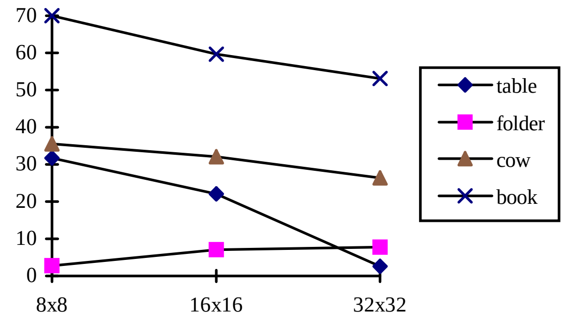


Figure 5. Mismatch ratios for the detected classes considering distinct number of cluster units for the neural network (average results).

in a sense that one more sample could make change the current weight configuration significantly. Using more than 15000 samples for the network training would produce configuration slightly different from the current state to the next. In this situation the neighborhood would be already reduced to include only the winning cluster unit and the gain factor could be reduced to have almost zero influence on the winning unit. This described situation was attained by parameters settings actually used, such as the number of nodes, decreasing factor for neighborhood size and gain factor, among others. These settings yielded good results. On the other hand, no extensively experiments were considered with lower decreasing ratios for neighborhood size and gain factor in a way that the influence of both neighborhood and gain factor could still be significant above 20000 or even more samples.

The new data set produced, i.e., the network weights, must then be clustered in order to obtain the classes. By applying the selected clustering method described in the last session, the classes in the network were recognized as clusters, and the cluster centers were determined. The chromaticity of each pixel in the image was assigned to the closest cluster center, resulting in the segmentation the image objects based on their color.

Error evaluation

A quantitative evaluation for the segmentation results was conducted considering a comparison with an human-assisted segmentation. This reference segmentation is showed in Figure 4, and was obtained manually. This segmentation was guided not only by the color attribute, but also considered the cognition of the objects (cow model, folder etc.). Based on this reference segmentation, the segmentation produced by the proposed method is compared pixel by pixel, evaluating the number of discrepancies between them. This coefficient, a mismatch ratio, is given by the number of correct class assignments over the total of points belonging to a class (manually determined), as in equation (7). M_i is the mismatch ratio, C_i is the number of correct classifications, and K_i is the total number of points that belong to class i.

$$M_i = \frac{C_i}{K_i} \tag{7}$$

The size, or number of nodes, of the network can vary. The adopted networks were structured as planar squared arrays. For network of size 4×4 and lesser the feature space tended not to converge, for various settings on neighborhood decreasing factor and gain function parameters. The obtained results for this specific situation included no topologic relationship among adjacent nodes in the network. In this regard, two considerations should be made: the number of samples were not enough and the convergence could occur with 40000 samples or more, or the decreasing factors for both neighborhood size and gain function values should decay more slowly (below a 0.01% reduction a sample). Obviously, the more the neighborhood delays to reach zero (including only the winning cluster unit), the higher the number of samples needed to be taken as input for the network. Some few experiments considering these situations were carried out with no significant results, but this situation has not been fully exploited.

The error estimate for distinct number of cluster units in the network are showed in Figure 5. The network sizes were rectangular arrays 8×8, 16×16, and 32×32. The displayed data were generated by the average results of experiments on networks with distinct initial configurations. Other parameters, such as initial weights and decreasing factors for neighborhood and gain function were maintained invariable, and the number of samples was about 2000. The initial values for the gain function was 0.30 at the winning cluster unit and 0.05 at the neighborhood border; the intermediate values were linearly interpolated. At each step (sample submission to the network for learning), the gain function for the winning node was reduced by a factor of 0.1%, while the neighborhood border value was left unchanged. When the gain function value for the winning node was reduced below the border value, both were left with equal values. Good results can be observed when the network had size 32×32, evidencing lower mismatch ratios for most of the detected classes. Higher sizes for the network did not presented better results, which were similar to those obtained with the 32×32 network. Only for the folder object the results were better for a small number of nodes.

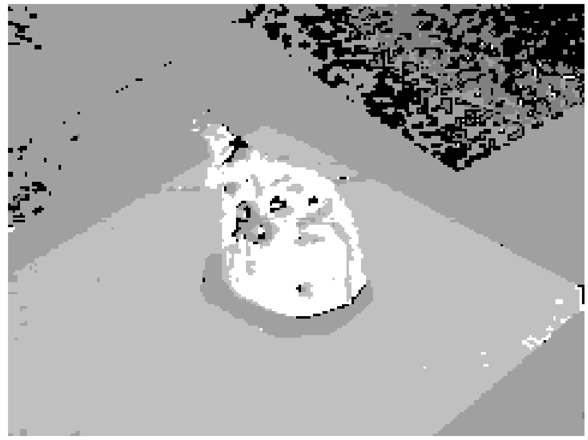
Additional considerations on neighborhood and gain function reduction factors

When the factor which reduces the gain function values (which are responsible by the influence of an input over the nodes of the network) is more than 1%, the gain function tends to zero very fast. When this factor is below 0.01, the convergence to zero is very slow.

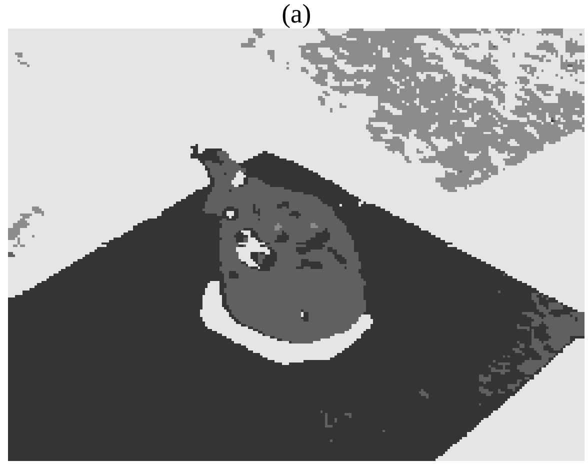
Similar comments can be made regarding the factor used to reduce the neighborhood size at each sample submission. For lower values (about 0.01 or lesser), the required number of samples was high, as it was for the gain function, and no improvement in segmentation could be noticed in this situation. In the opposite side, a “false contour” effect could be visually observed when the neighborhood radius were reduced by a factor higher than 1%. In other words, the reduction of the influence of a sample over the neurons in the specified neighborhood could be considered too coarse, with no smoothness. When this happens, the clustering procedure detected false clusters and the results were explicitly unreliable.

Segmentation results

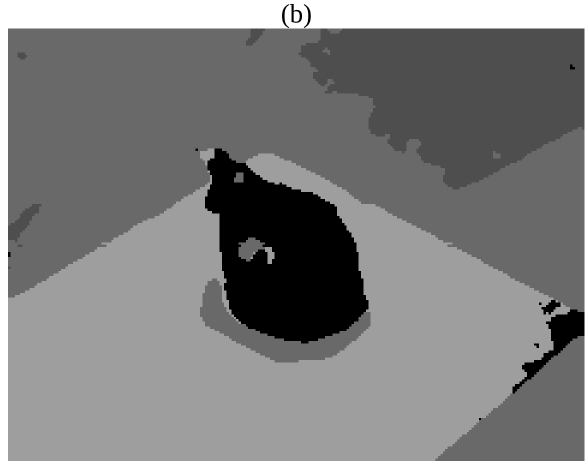
The results showed in Figure 6 represents segmentation results for some configurations. In Figure 6(a) the original image was segmented with a 32×32 neural network, and 3000 samples extracted from the image. The gain function started with a value of 0.30 of influence for the winning node and 0.05 for the neighborhood border; at each sample the value for the winning node was reduced



(a)



(b)



(c)

Figure 6. Obtained segmented images, with distinct gray levels to represent distinct classes (see text). (a) segmentation of the original image; (b) segmentation of a blurred version of the original image (Gaussian blur with variance 4); and (c) segmentation of a blurred version of the original image (Gaussian blur with variance 17). The grayscale among images do not match because they are automatically selected by segmentation program. See text for details.

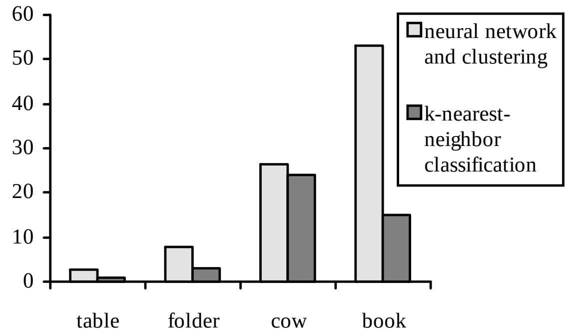


Figure 7. Comparison between the proposed method for segmentation and the supervised method proposed in [Moreira & Costa (1995)]. The vertical axis correspond to the mismatch ratio obtained in the classification. The values are an overall average considering the various parameters involved in both methods and should be considered more qualitative than quantitative.

by an amount of 0.1% of the current value. The same decreasing factor was used to reduce neighborhood size at each step. The mentioned problems during the image acquisition can be noted, but the overall correctness of the segmentation can be observed moreover. In Figure 6(b) the same process and parameters were used, but the original image was blurred by a Gaussian with variance 4. Since small details in the image were diminished, the final regions had a higher degree of smoothness. Finally, in Figure 6(c) the original image was heavily blurred with a Gaussian with variance 17 before segmentation, and the same parameters for the SOM and for clustering were employed. The perceived smoothness is even higher than in (b), but both details and region border were to a large extent lost. The neighborhood diameter started with about 80% of the side length of network array for all examples.

The results showed in Figure 6 entail a multiscale approach, which can be used to select the main features in a highly blurred version of the image. Therewith the process is refined by analysis of the "higher resolution" levels, defining more accurate borders for the segmented objects.

A comparison with other segmentation method

A comparison with the segmentation by a *k*-nearest-neighbor classification proposed by Moreira and Costa [Moreira & Costa (1995)] has also been considered. Since the same image was used for the experiments, the error rates can be directly compared. In Figure 7 the mismatch error for the objects considered are compared. The values are an average of the errors found for both methods, and must be analyzed more as a qualitative measure than a quantitative one. The main point to be observed is the better overall performance of the segmentation by the *k*-nearest-neighbor classification approach. However, the fact that the sample selection of this method is supervised, i.e., selected by an operator, must be taken into account. The segmentation using the SOM and its associated clustering approach detects classes without supervision. Sometimes, however, the number of clusters determined by the clustering procedure is overestimated. In this case, some homogeneous regions can be split in two or more classes. When this fact occurs, the mismatch ratio increases, reflecting that only one of such regions are considered as a specific object. The book object, which can be observed in Figure 6(a), has been considered by the technique as presenting as at least two classes (two distinct gray levels can be noted).

From a global point of view, the method shows an improvement over the *k*-nearest-neighbor approach in the sense that the task of segmentation can be achieved without supervision without major shortcomings. Better results should be achieved by considering alternative clustering techniques to determine the classes. This can consider either the LVQ (*learning vector quantization*) [Kangas et al. (1990)] or the GLVQ (*generalized learning vector quantization*) [Pal et al. (1993)], among others. These approaches could lower down the mismatch ratios, improving the classification results. Similar comments must be made about the multiscale approach, which deals with a coarse-to-fine procedure, which should refine the final segmentation.

Conclusions

This paper has presented a segmentation method for color images based on the chromaticities of the objects. The segmentation makes use of a Kohonen network (which implements the so called self-organizing map), which was able to discriminate the main chromaticities of the image. Sample points are taken from the image (already normalized) and submitted to the network in its learning phase. When the feature map has been formed, i.e., tended to a stable state, the present features (the main chromaticities) are clustered to detect not only the number of significant clusters but also their centers. Then it is possible to classify the pixels in the image assigning them to the closest cluster center. The final segmentation is composed by the classified pixels, which were assigned to one of the determined classes.

The results substantiate the feasibility of the method, which is able do detect the classes in a coherent manner. Also, the mismatch error measures allowed to evaluate the method as presenting a reasonable reliability.

The investigation in this subject are not concluded yet. The clustering phase must be improved with the investigation of other techniques which lead to minimize an error measure in-between clusters. Such techniques would involve LVQ and GLVQ, among other classic clustering approaches. Investigations on gain function behavior, other than the adopted linear interpolation used, must also be considered, as well as a multiscale approach.

References

Anderson, J. A. *An introduction to neural networks*. MIT Press, 1995.

Celenk, M. A color clustering technique for image segmentation. *Comp. Vis. Graph. Image Process.*, *52*:145–170, 1990.

Coleman, G. B. and H. C. Andrews. Image segmentation by clustering. *Proc. IEEE*, 67(5): 773-785, 1979.

Costa, L. da F.; V. O. Roda; and R. Köberle. A biologically-inspired system for visual pattern recognition. *IEEE International Symposium on Industrial Electronics*, Santiago-Chile, 1994.

Cumani, A. Edge detection in multispectral images. *CVGIP Graph. Models and Im. Process.*, *53*:40–51, 1991.

Duda, R. O. and P. E. Hart. *Pattern classification and scene analysis*. John Wiley and Sons, Inc., New York, 1973.

Godfrey, K. R. L. and Y. Attikiouzel. Applying neural networks to colour image data compression. IEEE Region 10 Conference, Tencon 92. Melbourne, Australia, 1992.

Hertz, J.; A. Krogh and R. G. Palmer. *Introduction to the theory of neural computation*. Addison-Wesley Publ. Co., 1991.

Huntsberger, T. L.; C. L. Jacobs; and R. L Cannon. Iterative fuzzy image segmentation. *Patt. Recogn.*, *18*:131–138, 1985.

Huntsberger, T. L .; C. Rangarajan; and S. N. Jayaramamurthy. Representation of uncertainty in computer vision using fuzzy sets. *IEEE Trans. Computers*, *35*:145–156, 1986.

Kangas, J. A.; T. K. Kohonen; and J. T. Laaksonen. Variants of self-organizing maps. *IEEE Trans. Neural Networks*, 1(1):93-99, 1990.

Kohonen, T. *Self-organization and associative memory*. Springer-Verlag, Berlin, 1989.

Lee, H.-C. and D. R. Cok. Detecting boundaries in a vector field. IEEE *Trans. Sign. Process.*, *39*:1181–1194, 1991.

Levine, M. D. *Vision in man and machine*. McGraw Hill Publ. Co., N. Y., 1985.

Lim, Y. W. and S. U. Lee. On the color image segmentation algorithm based on the thresholding and the fuzzy *c*-means techniques. *Patt. Recog.*, *23*:935–952, 1990.

Moreira, J. and L. da F. Costa. Multispectral image segmentation by chromaticity classification. *VIII SIBGRAPI*, pp. 119-125. São Carlos, 1995.

Ohlander, R.; K. Price; and D. R. Reddy. Picture segmentation using a recursive region splitting method. *Comp. Graph. Image Process.*, *8*:313–333, 1978.

Ohta, Y.-I.; T. Kanade; and T. Sakai. Color information for region segmentation. *Computer Graphics and Image Processing*, *13*:222–241, 1980.

Pal, N. R.; J. C. Bezdec; and E. C.-K. Tsao. Generalized clustering networks and Kohonen's self-organizing scheme. *IEEE Trans. Neural Networks*, 4(4): 549- 557, 1993.

Pratt, W. K.. *Digital image processing*. John Wiley & Sons, Inc., 2nd ed., 1991.

Schlünzen, E. T. M. *et al*. Classificação de dados multiespectrais utilizando redes neurais: a influência da amostragem no processo de treinamento. *Anais do Workshop sobre Visão Cibernética*. São Carlos, Agosto 1994.

Trivedi, M. M. and J. C. Bezdek. Low-level segmentation of aerial images with fuzzy clustering. *IEEE Trans. Syst. Man Cybern*, *16*:589–598, 1986.

Yin, H. and N. M. Allinson. On the distribution and convergence of feature space in self-organizing maps. *Neural Computation*, 7:1178-1187, 1995.