1994 International Symposium on Speech, Image Processing and Neural Networks, 13-16 April 1994, Hong Kong

PARALLEL TEXTURE ANALYSIS USING HIGH-ORDER STATISTICS AND NEURAL NETWORKS

Helder Araújo

Institute of Systems and Robotics
Department of Electrical Engineering
University of Coimbra
3000 Coimbra, Portugal
email: helder@uc.pt

Jorge Dias

Institute of Systems and Robotics
Department of Electrical Engineering
University of Coimbra
3000 Coimbra, Portugal

Anibal T. Almeida

Institute of Systems and Robotics
Department of Electrical Engineering
University of Coimbra
3000 Coimbra, Portugal

Abstract

In this paper we present a texture analysis and classification method based on the computation of high-order statistics of gray-levels or selected features. The computation of these high-order statistics can be performed in parallel in non-overlapping windows that cover the full image. Classification is performed by neural networks. In this extended summary results are presented for back-propagation networks only, but several other types of neural nets were also tested.

1. Introduction

Texture is a visual quality that has been extensively researched in computer vision and psychophysics. Texture is generally defined as a structure which is made up of a large ensemble of similar elements or patterns without one of these drawing special attention [1]. The observer gets a global impression of uniformity when he looks at a texture. Textures can be classified in between two extreme classes: deterministic and stochastic. Deterministic textures are described by primitives and placement rules that define the primitives' spatial distribution. In stochastic textures it is not possible to recognize a subpattern or dominant repetition frequency. Stochastic textures can be considered as a result of a 2D stochastic process.However most of natural textures do not fall into one of these two categories, and are best described as a "mixed" type.

Approaches for texture analysis, classification and synthesis can also be divided into two main categories: structural and statistical [2]. This division is in correspondence with the above described categorization of textures.Statistical approaches use statistical measures of the gray level distributions or of selected features to describe textures.Structural methods characterize textures by describing their primitives and their placement rules [3].

2. The Basis Of The Method

The method we present in this paper is based on the estimation of high-order statistics (higher than third-order) of the gray levels or of selected local features. Many statistical methods were proposed based on Julesz findings that the human visual system uses global first- and second-order statistics for texture discrimination [4, 5]. One category of such methods are those based on the computation of the cooccurrence matrices [6, 7].Other methods were proposed that use third-order statistics as well as first- and second-order statistics. The use of third-order statistics enabled the extraction of more of the structural and directional information of the texture.

The use of second-order moments is equivalent to the use of the autocorrelation function. In many cases this is a natural choice for texture characterization and the information extracted from the second-order statistics is sufficient for its analysis and classification. However it is well known that the information contained in the autocorrelation function is only sufficient for a full description of signals in the cases of Gaussianity. In the cases where deviations from Gaussianity and phase relations occur higher order statistics (also known as "cumulants") have to be used. The use of high-order statistics enables also the detection and characterization of nonlinear properties in signals [8].

Most of the real world signals are non-Gaussian and thus have non-zero higher-order spectra. For texture classification distinct classification features can be extracted from the higher-order spectra. In this work we actually used estimates of the probability density functions and corresponding moments to classify textures. The computation of the estimates of the high-order probability density functions is performed in non-overlapping windows defined in each image.

3. The Model

Let us assume that we have C texture classes $(w_1, w_2, ..., w_C)$. Let $x_1, x_2, ..., x_d$ be the values that a specific feature takes in d points of a two-dimensional grid. These feature values can be, for example, the gray level values of d pixels. Let us also assume that we are using $k * l$ (k lines, l columns) images.This two-dimensional grid is a $r * s$ window defined within the image, with $r << k$ and $s << l$. Let **X** be a d-dimensional vector of the form

$$\mathbf{X} = (x_1, x_2, ..., x_d)$$

where the x_i are the values of the feature in specific points of the 2D grid. One important question that has to be taken into account is the spatial distribution of the d points. These d points have to be uniformly distributed all over the $r * s$ window. This is to avoid that local structures determine the probability density functions to be estimated. For the classification we need to have estimates of the probability density functions

$$p (\mathbf{X} / w_i) , \quad i = 1, 2, ..., C$$

In the cases where $d = 2$ these estimates can be approximated by the cooccurrence matrices. Once these probability density functions are estimated classification can be performed in a number of different ways [9]. One can, for example, use the Bayes decision rule to classify tan unknown texture. Given a vector **X** we want to find out a texture class w_i for which the probability

$$p (w_i / \mathbf{X}) , \quad i=1, 2, ..., C \qquad \text{is maximum.}$$

According to the Bayes rule

$$p(w_i / \mathbf{X}) = \frac{p(\mathbf{X} / w_i) * p(w_i)}{p(\mathbf{X})}$$

However in the work described in this paper we used a different approach. To characterize each texture class we compute first-, second- and third-order moments and entropies of the estimates of the probability density functions $p (\mathbf{X}/ w_i)$. Each element x_i of the vector **X** is quantized and assumes a finite number of different values.For example in the case where the feature used is the pixel's gray level value the x_i will assume a maximum of 256 different values (for 8 bits/pixel images). Each x_i will be represented as a vector itself. We can write

$$x_i = (x_{i1}, x_{i2} , ..., x_{it})$$

As **X** is a multidimensional vector there are several first-, second- and third-order moments. For example we can define several means:

$$\mu_{x_i} = \sum_{j=1}^{j=t} x_{ij} \left(\sum_{1k=1}^{1k=t} \sum_{2l=1}^{2l=t} \cdots \sum_{\substack{am=1 \\ a \neq i}}^{am=t} \cdots \sum_{dn=1}^{dn=t} p(x_{1k}, \ldots, x_{dn}) \right)$$

The values calculated are used as descriptors or features of the texture classes. For classification we used neural networks (we used several types of networks and compared their performances). The neural net is trained with these descriptors. For recognition we will also use moments of the multidimensional probability distribution p (**X**). In order to obtain an estimate of the probability density function which is reasonably accurate we need to have a reasonable number of samples. That is achieved requiring that the window where the spatial sampling of the d pixels is performed be of dimensions significantly smaller than the image. This way each image or subimage provides us with a sufficient number of samples. The underlying assumption is that all the analyzed patches can be considered as representing samples of the texture field. We call this method parallel because the computation of the estimates of the probability density functions can be performed in parallel. Indeed for that purpose the image is divided into several non-overlapping windows of dimension $r *s$. These samples can be taken into account (in the computation of the estimates) simultaneously.

4. Implementation

One problem with the implementation of this model is the high-dimensionality of the estimates p(**X**/w_i). For example, if we are dealing with images with 256 gray levels the estimate of a third-order probability density function would require an impracticable amount of memory. There are several alternatives to solve this problem and we chose one solution that somehow "preserves" the direct computation of high-order statistics. For that purpose we used a kind of gray level sampling.

Let us assume we are dealing with 8 bits/pixel images. From each image we generate a number of binary images. This number depends on the resolution of the gray level sampling we want to perform. In our case we generated from each image four binary images. We used 4 gray level thresholds equally spaced in the range [0, 255]. The thresholds we used were 64, 128, 192, 224.

Let $f(x,y)$ be the gray level of pixel (x, y). Let T be the value of the threshold. The binary images $g(x,y)$ were generated by applying the condition:

$$g(x,y) = \begin{cases} 1, & \text{if } f(x,y) \geq T \\ 0, & \text{if } f(x,y) < T \end{cases}$$

In each one of these binary images we compute the histograms $p (\mathbf{X}/w_i)$. The use of binary images enabled us to compute high-order statistics. In this case (binary images) the estimate of the probability density function $p(X/w_i)$ is a unidimensional vector (as a result of some manipulation of the pixels' binary values). We typically used 10 pixels which means that we were computing tenth-order statistics. These histograms were computed in 16*16 windows. From each histogram the mean, variance, third-order moment and entropy were computed. Therefore each texture class was represented by a set of 16 values (4 statistical parameters per binary image). These measures were used as inputs of the neural networks.

The types of neural nets tested with this type of statistical measures were back-propagation, learning vector quantization, probabilistic neural nets, self-organizing maps and counter-propagation. However in this paper only the results obtained with back-propagation are presented.

5.Results

The tests were performed on images from the album [10]. Eight classes were considered: grass lawn (D9), cloth (D19), beach sand (D29), water (D38), wood (D68), raffia (D84), pigskin (D92) and fur (D93).The training set was composed by 24 images from each class and the test set was composed by 32 images from each class. All the images were digitized by performing rotations, translations and variations of illumination on the textures' photographic plates.

The results presented here are for a back-propagation net with one-hidden layer. The records in the training file were presented randomly, whereas the test file was presented sequentially. The layer parameters used were those described in Table 1.

Table 1: Layer parameters

Layer	*Summation*	*Transfer*	*Output*	*Learning rule*	*Learning schedule*
input	sum	linear	direct	none	none
hidden	sum	tanh	direct	cumulative epoch = 16	hidden1 (see below)
output	sum	tanh	direct	cumulative epoch = 16	out (see below)

Each layer was fully connected to the next one. We used different values for the learning coefficients in the different layers and they were modified as the learning progressed.

Table 2: Coefficients used in the hidden layer

Learning step	0 --> 10000	10000 --> 30000
Gain term	0.3	0.15
Momentum term	0.4	0.2

The learning coefficients used in the hidden layer are described in Table 2. The coefficients used in the output layer are described in Table 3.

Table 3: Coefficients used in the output layer

Learning step	0 --> 10000	10000 --> 30000
Gain term	0.15	0.075
Momentum term	0.4	0.2

The results for the network with different number of nodes at the different layers are presented in Table 4.

Table 4: Performance as a function of the number of nodes

Type of net	*Performance*	
	Learn 15000	*Learn 30000*
Back-propagation 16-4-16	208/256	216/256
Back-propagation 16-6-16	232/256	240/256
Back-propagation 16-8-16	248/256	248/256
Back-propagation 16-10-16	256/256	256/256
Back-propagation 16-12-16	256/256	256/256
Back-propagation 16-14-16	256/256	256/256
Back-propagation 16-16-16	256/256	256/256

In the sequence of numbers 16-4-16, the first number represents the number of nodes in the input layer, the second number represents the number of nodes in the hidden layer and the third number represents the number of nodes in the output layer.

Learn 15000 means that 15000 records were presented to the network during the training phase. The number in the performance columns represent the number of images correctly classified out of the total of 256 images. The results for a one-hidden layer back-propagation network with epoch size variation are described in Table 5.

Table 5: Performance as a function of epoch size variation

Back-propagation 16-8-16	*Performance (learn 15000)*
Epoch = 1	240/256
Epoch = 4	248/256
Epoch = 8	248/256
Epoch = 16	248/256
Epoch = 24	248/256
Epoch = 32	232/256
Epoch = 192	176/256

6. Conclusions

The new method for texture analysis and classification described in this paper is based on the parallel computation of features. These features are estimates of high-order probability distributions. Based on these estimates of probability distributions a set of descriptors (which are several moments of the distributions as well as their entropy measures) is computed. These descriptors were then used with back-propagation neural networks to perform texture classification. As expected (due to the high dimensionality of the estimated probability distributions) the method proved to be quite robust against translations, rotations and variations of illumination.

References

[1] L. V. Gool, P. Dewaele, A. Osterlinck, "Texture Analysis Anno 1983", *Computer Vision, Graph. and Image Processing*, Vol. 29, 1985.

[2] R. M. Haralick, "Statistical and Structural Approaches to Texture", *Proceedings of the IEEE*, Vol. 67, No. 5, pp. 786-804, 1979.

[3] F. M. Vilnrotter, R. Nevatia, K. E. Price, "Structural Analysis of Natural Textures", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 1, 1986.

[4] B. Julesz, "Visual Pattern Discrimination", *IRE Trans. on Information Theory*, Vol. 8, No. 2, February 1962.

[5] B. Julesz, "Experiments in the Visual Perception of Texture", *Scientific American*, vol. 232, No. 4, April 1975.

[6] R. M. Haralick, K. Shanmugan, I. Dinstein, "Textural Features for Image Classification", *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC-3, No. 6, 1973.

[7] L. S. Davis, M. Clearman, J. K. Aggarwal, "An Empirical Evaluation of Generalized Cooccurrence Matrices", *IEEE Trans. on Pattern Analysis and machine Intelligence*, Vol. PAMI-3, No. 2, 1981.

[8] M. Tsatsanis, G. Giannakis, "Object and Texture Classification Using Higher Order Statistics", *IEEE Trans. on Pattern Analysis and machine Intelligence*, Vol. PAMI-14, No. 7, pp. 733-750, July 1992.

[9] H. Araujo, A. T. Almeida,"A parallel architecture for texture analysis based on the concept of associative memory", *Industrial Metrology - The International Journal of Automated Measurement and Control*, Vol.1, pp. 127-138, Elsevier, 1990.

[10] P. Brodatz, *Textures- A Photographic Album for Artists and Designers*. New York: Dover, 1966.