FACTA UNIVERSITATIS (NIŠ)
Series: Electronics and Energetics vol. 12, No.1 (1999), 103-119

UDC 621.391

ON TWO-DIMENSIONAL SHAPE RECOGNITION USING MOMENT INVARIANTS AND AR MODELS

Milan Marković, Predrag Pejnović and Srdjan Stanković

Abstract. This paper is dedicated to the problem of two-dimensional (2D) shape recognition from the point of view of possible optimization regarding feature extraction and classification methods. Moment invariants and stochastic AR models are considered as feature extraction methods. For classification, we analyze performance of the Bayesian parametric and nonparametric classifiers, as well as of the multilayer perceptron, applied as a nonparametric classifier. Experimental analysis is based on real data. Evaluation of the considered methods is done on the basis of the Bayes error estimates, calculated on the corresponding data sets.

1. Introduction

In the computer vision theory, the problem of 2D shape analysis and recognition has been treated either structurally or analytically [4]. The areas of application of 2D shape recognition include: classification of blood cells, chromosomes, corn kernels, industrial inspection, target recognition [5], scene analysis [4] and modeling of biological systems [15]. The methods proposed for 2D shape description include either statistical approaches based on the method of moments [1],[5],[7],[9],[12],[16], or the approaches based on some functions derived from the boundary using Fourier analysis [18], autoregressive (AR) models [8],[3],[2],[14] and wavelet transform [17]. In this paper, we experimentally analyze two feature extraction methods: moment invariants and stochastic AR models. Namely, two different groups

Manuscript received January 12, 1999.

M. Marković is with Institute of Applied Mathematics and Electronics, Kneza Miloša 37, 11000 Belgarde, Yugoslavia, e-mail: markovic@kiklop.etf.bg.ac.yu. P. Pejnović is with S.M.T. d.o.o., KCS-Poliklinika, Višegradska 26, 11000 Belgrade, Yugoslavia, e-mail: pejnovic@eunet.yu. S.Stanković is with Faculty of Electrical Engineering, Bulevar Revolucije 73, 11000 Belgrade, Yugoslavia, e-mail: stankovic@etf.bg.ac.yu.

of moment invariants and two different types of AR models are considered. In addition, this work is dedicated to the analysis of possible applications of Bayesian parametric (linear and quadratic) and nonparametric ($k-NN$) classifiers, as well as of the multilayer perceptron (MLP), applied as a non–parametric pattern classifier, in the recognition of 2D shapes modeled by the moments and AR models. A comparative experimental analysis is done on the basis of real objects photographed by TV camera. Gray level images are tresholded to obtain binary silhouettes. Contours are obtained by using an original algorithm for contour extraction. In the case of moment invariants, both silhouettes and contours are used for extracting feature vectors [12]. For the AR feature extraction method, parameter estimates are obtained by using the non–recursive least square method (LS) [10], on the basis of the coordinates of the given contour elements. The obtained parameter estimates are used as feature vectors that are to be classified. The classification errors, obtained by using the considered classifiers and given feature extraction methods are evaluated by using the resubstitution method [6]. The reason for using this method is a small number of available samples of the real object shapes. Performance of the classifiers is analyzed referred to the appropriate Bayes error estimates obtained by a very accurate procedure combining the resubstitution and the leave–one–out methods [6].

This work is organized as follows. The main principles of the moment invariants and AR modeling of 2D shapes, are described in Section 2 and 3, respectively. Main characteristics of the Bayesian classifiers and the multilayer perceptron are briefly described in Section 4. A comparative experimental analysis is presented in Section 5, while the conclusion is given in Section 6.

2. Moments

In this paper, two feature sets, based on central moments, are considered: moment invariants [5], and Zernike moments [9]. Namely, Hu [7] introduced seven nonlinear functions defined on regular moments which are translation, scale and rotation invariant. These seven, so called moment invariants, expressed via normalized central moments, Eq. (1), are used in a number of pattern recognition problems [5],[12]. Teaque [16] has suggested the application of orthogonal moments based on the theory of orthogonal polynomials to efficient computation and image recovery. A class of such orthogonal moments possessing the rotation invariance property are Zernike moments [9]. Rotating the image does not change the magnitude of its Zernike moments and they are used as rotation invariants for feature extraction. These features can easily be computed up to an arbitrary high

order. Another important property of Zernike moments is the ease of image reconstruction from them [9]. To obtain scale and translation invariance, the image is first subjected to a normalization process using its central moments. The rotation invariant Zernike features are then extracted from the scale and translation normalized image. Teaque [16] has shown relations among moment invariants, invariant Zernike features and central moments. To extract features from binary images, where 2D stochastic function of contour points $f(x_i, y_i)$ is either 1 or 0, central moments of $(p+q)$ order are expressed by

$$\mu_{pq} = \frac{1}{N} \sum_{i=1}^{N} f(x_i, y_i)(y_i - y)^p (x_i - x)^q, \tag{1}$$

where (x, y) are coordinates of the central point of the contour, and N is the number of pixels. The central moments are normalized by the factor $m = \mu_{00}^{(p+q+2)/2}$. This normalization corresponds to having the total image power always equal to unity. In this paper, a set composed of 11 Zernike moment invariants expressed via central moments up to the 4th order $(p+q \leq 4)$, Eqs. (2) and (3) are used to obtain the feature vectors of 11 and 12 features [12].

$$\begin{aligned}
MI_1 =& \frac{3}{\pi}(2(\mu_{20} + \mu_{02}) - 1) \\
MI_2 =& \frac{9}{\pi^2}((\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2) \\
MI_3 =& \frac{16}{\pi^2}((\mu_{03} - 3\mu_{21})^2 + (3\mu_{12} - \mu_{30})^2) \\
MI_4 =& \frac{144}{\pi^2}((\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2) \\
MI_5 =& \frac{13824}{\pi^4}((\mu_{03} - 3\mu_{21})(\mu_{03} + \mu_{21})(\mu_{03} + \mu_{21})^2 - 3(\mu_{30} + \mu_{12})^2 \\
& -(\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12}) + (\mu_{30} + \mu_{12})^2 - 3(\mu_{03} + \mu_{21})^2) \\
MI_6 =& \frac{864}{\pi^3}((\mu_{02} - \mu_{20})(\mu_{03} + \mu_{21})^2(\mu_{30} + \mu_{12})^2 \\
& +4\mu_{11}(\mu_{03} + \mu_{21})(\mu_{30} + \mu_{12})) \\
MI_7 =& \frac{25}{\pi^2}((\mu_{40} - 6\mu_{22} + \mu_{04})^2 + 16(\mu_{31} - \mu_{13})^2) \\
MI_8 =& \frac{25}{\pi^2}(4(\mu_{04} - \mu_{40}) + 3(\mu_{20} - \mu_{02}) + 4(4(\mu_{31} + \mu_{13}) - 3\mu_{11})^2) \\
MI_9 =& \frac{5}{\pi}(6(\mu_{40} + 2\mu_{22} + \mu_{04}) - 6(\mu_{20} + \mu_{02}) + 1)
\end{aligned} \tag{2}$$

$$
\begin{aligned}
MI_{10} =& \frac{250}{\pi^3}((\mu_{40} - 6\mu_{22} + \mu_{40})(4((\mu_{04} - \mu_{40}) + 3(\mu_{20} - \mu_{02})^2) \\
& -4(4(\mu_{31} + \mu_{13}) - 3\mu_{11})^2) - 16(4(\mu_{04} - \mu_{40}) \\
& +3(\mu_{20} - \mu_{02}))(4(\mu_{31} + \mu_{13}) - 3\mu_{11})(\mu_{31} - \mu_{13})) \qquad (3) \\
MI_{11} =& \frac{30}{\pi^2}(4(\mu_{04} - \mu_{40}) + 3(\mu_{20} - \mu_{02})(\mu_{02} - \mu_{20}) \\
& +4\mu_{11}(4(\mu_{31} + \mu_{13}) - 3\mu_{11}))
\end{aligned}
$$

The 11–dimensional feature vectors represent Zernike moment invariants (Eqs. (2) and (3)) calculated for a silhouette. In order to take advantage of the information content of both the boundary and the silhouette of a 2D shape, two sets of the first six Zernike moment invariants are computed, Eqs. (2) and (3), (one set derived from the boundary and the other from the silhouette). Applying this technique, 12–dimensional feature vectors are obtained.

3. AR modeling of 2D contours

AR modeling applied to the classification of 2D shapes has attracted the attention of researchers in the field [8],[3],[2],[14]. The AR model of the contour of a 2D shape can be, in general, represented as

$$x_t = \alpha + \sum_{j=1}^{n} \theta_j x_{t-j} + \beta\omega_t, \qquad (4)$$

where x_t describes a point on the contour, θ_j $(j = 1, \ldots, n)$, α and β represent the unknown parameters and ω_t is a zero-mean unit variance white noise sequence. Depending on the adopted approach, x_t can be either the radial distance from a point inside the contour [8] or the $x - y$ coordinate vector [3],[2] possibly expressed as a complex number [14]. Accordingly, we shall consider the following two types of AR models:

One–dimensional (1D) AR model. Parameters of the 1D AR model are estimated from the samples of the boundary sequence using the conventional LS method [10]. The LS estimates can be obtained as follows. Define φ as "the regression vector", which consists of radii samples of boundary points (distance between the contour center and a boundary point), i.e.

$$\phi(t) = [-x(t-1) - x(t-2) - \cdots - x(t-n)]^T \qquad (5)$$

the autocorrelation matrix

$$\begin{aligned} R(N) =& \frac{1}{N}\sum_{t=1}^{N}\phi(t)\phi^T(t) \\ [R(N)]_{ij} =& \frac{1}{N}\sum_{t=1}^{N}x(t-i)x(t-j); \quad i \geq 1, \quad j \leq n \end{aligned} \tag{6}$$

and

$$f(N) = \frac{1}{N}\sum_{t=1}^{N}\phi(t)x(t). \tag{7}$$

The one-step predictor for 1D model can be written as $\hat{x} = \phi^T\theta$ (linear regression), where $\theta = [\theta_1, \ldots, \theta_n]$ is the parameter vector. The criterion to be minimized is defined as the sum of squares of the prediction errors

$$e = \frac{1}{N}\sum_{k=1}^{N}|x(k) - \hat{x}(k)|^2. \tag{8}$$

The LS estimate of the parameter vector θ is given by

$$\hat{\theta}_{LS} = R^{-1}(N)f(N). \tag{9}$$

In 1D case, the feature vectors consist of n features $(\theta_1, \theta_2, \ldots, \theta_n)$. Namely, the number of features is equal to the order of the applied AR model.

Two–dimensional (2D) AR model. The 2D AR model parameters are estimated from the samples of x, y coordinates of the boundary sequence by using the LS method, as described for 1D model. In this case, the parameters θ_j $(j = 1, \ldots, n)$ are, in fact, 2×2 coefficient matrices, given by

$$\theta_j = \begin{bmatrix} \vartheta_{1j} & \rho_{1j} \\ \vartheta_{2j} & \rho_{2j} \end{bmatrix} \tag{10}$$

and a 2×1 process mean vector, α, i.e. $E[x(k)] = \alpha$. The process mean vector α and the model residual error covariance matrix, β, can be estimated from

$$\begin{aligned} \hat{\alpha} =& \frac{1}{N}\sum_{k=1}^{N}\begin{bmatrix} x_k \\ y_k \end{bmatrix} \\ \hat{\beta} =& \frac{1}{N}\sum_{k=1}^{N}\left[\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \hat{\theta}^T\varphi\right]\left[\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \hat{\theta}^T\varphi\right]^T \end{aligned} \tag{11}$$

In this case

$$\theta = [\,\theta_1 \quad \theta_2 \quad \cdots \quad \theta_n\,]^T$$

$$\varphi(t) = \left[- \begin{bmatrix} x(t-1) \\ y(t-1) \end{bmatrix}^T - \begin{bmatrix} x(t-2) \\ y(t-2) \end{bmatrix}^T - \cdots - \begin{bmatrix} x(t-n) \\ y(t-n) \end{bmatrix}^T \right]^T$$

$$R(N) = \frac{1}{N} \sum_{t=1}^{N} \varphi(t)\varphi^T(t)$$

$$f(N) = \frac{1}{N} \sum_{t=1}^{N} \varphi(t) \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}^T$$

For given $\varphi(t)$, $R(N)$, and $f(N)$, the LS estimate of the parameter vector is given by (9). The feature vector for classification consists of the following $(2n+2)$ elements [2], [14]:

- $2n$ elements obtained as the sum of squares of ϑ and ρ elements of the coefficient matrices θ_j ,i.e. $2n$ elements are

$$\{\sigma_{xj} = \vartheta_{1j}^2 + \vartheta_{2j}^2, \quad \sigma_{yj} = \rho_{1j}^2 + \rho_{2j}^2, \quad j = 1, \ldots, n\}$$

- An element defined as a combination of the mean vector α and the covariance matrix β: $\tau = \alpha^T \beta^{-1} \alpha$;
- An element defined as the sum of the eigenvalues of the estimated system matrix A_s when the AR model is written in the state-space form [2], [3], i.e.:

$$A_s = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \cdots & \theta_{n-1} & \theta_n \\ I_2 & 0 & 0 & & 0 & 0 \\ 0 & I_2 & 0 & & 0 & 0 \\ 0 & 0 & I_2 & & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & \cdots & I_2 & 0 \end{bmatrix} \tag{12}$$

4. 2D shape classification

Bayesian classifier. Bayesian classifier can be described by follows. Let us consider c classes of the training data set, ω_i, $i = 1, \ldots, c$; described by a posteriori probability functions $P(\omega_i|X)$. Bayes rule can be expressed as follows, [4]

$$P(\omega_i|X) = \frac{P(\omega_i)p(X|\omega_i)}{p(X)} \tag{13}$$

where $p(X)$ is the probability density function (pdf) of the sample X, $p(X|\omega_i)$ is the conditionally probability density function (cpdf), and $P(\omega_i)$ is the a priori probability of the class ω_i. Bayesian classifiers is based on the *Bayes decision rule:* sample X is classified in the class ω satisfying

$$P(\omega|X) = \max_{i \le i \le c} \{P(\omega_i|X)\}. \tag{14}$$

The Bayesian classifier, described by the equation (14) can be used for classification under one of the following three conditions:

1. In case when cpdfs of all the training data set classes are completely known.
2. With the assumption that cpdfs of the training data set classes belong to the parametric distribution families with known functional forms which are described by the finite number of parameters that should be estimated.
3. In case when the cpdfs of the training data set classes are estimated by using some of the nonparametric procedures.

In the first two cases we obtain *parametric* classifiers while in the third case we obtain nonparametric ones.

Parametric classifiers. In this paper, we assume that cpdfs of the training data set classes belong to the Gaussian distribution class, with parameters to be estimated on the basis of training data. The Gaussian distribution is completely described by only two parameters: *mean vector* (M) and *covariance matrix* (Σ). Multidimensional Gaussian distribution is described by the following equation

$$p(X) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(X-M)^T \Sigma^{-1} (X-M)\right] \tag{15}$$

Quadratic classifier is the optimal Bayesian classifier in the case of the classification of Gaussian data [4],[6]. The quadratic discrimination function has the following form

$$\begin{aligned} g_i(X) =& X^T \Sigma_i^{-1} X - 2 M_i^T \Sigma_i^{-1} X + M_i^T \Sigma_i^{-1} M_i \\ &+ \ln |\Sigma_i| - 2 \ln P(\omega_i) \end{aligned} \tag{16}$$

An arbitrary sample X is classified into the class ω_i which gives the minimum value of the discrimination function (16).

Linear classifier represents the Bayesian classifier of the Gaussian data with the assumption that all the classes of the training data set have the same covariance matrix. The linear discrimination function can be described as follows

$$g_i(X) = M_i^T \Sigma^{-1} X - \frac{1}{2} M_i^T \Sigma^{-1} M_i + \ln P(\omega_i) \tag{17}$$

Classification is done by choosing the class with maximum value of the discrimination function (17). We consider two versions of the linear classifier: *piecewise* and *pairwise* linear classifier.

Piecewise linear classifier. In this version, we use *a common covariance matrix*, Σ, for all training data set classes, obtained by the following equation

$$\Sigma = \sum_{i=1}^{c} P(\omega_i) \Sigma_i \tag{18}$$

Pairwise linear classifier. In the multi-class case, instead of using the common covariance matrix for all the classes, we use *the pairwise common covariance matrix, Σ_{ij}, for each pair of the training data set classes i and j*, obtained by the following equation

$$\Sigma_{ij} = \frac{P(\omega_i)\Sigma_i + P(\omega_j)\Sigma_j}{P(\omega_i) + P(\omega_j)} \tag{19}$$

Nonparametric classifiers. A nonparametric classifier does not rely on any assumption concerning the structure of the underlying density function. Therefore, the classifier becomes the *Bayesian classifier* if the density estimates converge to the true densities when an infinite number of samples is used. The resulting error is the *Bayes error*, the smallest achievable error given the underlying distributions. The Bayesian error is a very important parameter in pattern recognition, assessing the classifiability of the data and measuring the discrimination capabilities of the features even before considering what type of classifier should be designed.

k-NN classifiers are based on the nonparametric estimates of the cpdfs by using the $k - NN$ approach, which has the form [4],[6]

$$\hat{p}(\omega_i|X) = \frac{k-1}{N_i \nu_i(X)} \tag{20}$$

where k is the number of nearest neighbors to the sample X from the class ω_i of the training data set, N_i is the number of vectors in the class ω_i, and $\nu(X)$

is the volume of the set that accompany k nearest neighbors to the sample X from the class ω_i. The formulae for the volume calculation depend of the distance measure that is used. In this paper, the Mahalanobis distance is used [6]. Depending of what is assumed to have a fixed value for all of the training data set classes, either the number of k nearest neighbors or the volume $\nu(X)$, we obtain two types of the $k-NN$ classifiers: "voting" or "volumetric" $k-NN$ classifier, respectively.

"Voting" $k-NN$ classifier. In this case, by specifying the entire number k of nearest neighbors to the sample X we specify, in fact, the same volume $\nu(X)$ for all of the training data set classes. The sample X is then classified to the class with the largest number of nearest neighbors of the entire number k. In other words, the classification is done by choosing the maximum value of the discrimination function, given by [6]

$$g_i(X) = k_i, \quad \sum_{i=1}^{c} k_i = k \tag{21}$$

"Volumetric" $k-NN$ classifier. In this case, we assume the fixed number of nearest neighbors for each of the training data set classes. The estimates, given by (20), differ only by the volume $\nu_i(X)$ of each class. Including (20) in the likelihood ratio classifier term that is derived from the two–class Bayes decision rule [6]

$$-\ln\frac{\hat{p}(X|\omega_1)}{\hat{p}(X|\omega_2)} \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \tau - \ln\frac{P(\omega_2)}{P(\omega_1)} = \tau \tag{22}$$

we obtain the following inequality for "volumetric" $k-NN$ classifier in two–class case

$$-\ln\frac{(k-1)N_2\nu_2(X)}{(k-1)N_1\nu_1(X)} = -n\ln\frac{d_2(X_{kNN}^{(2)},X)}{d_1(X_{kNN}^{(1)},X)} - \ln\frac{N_2|\Sigma_2|^{\frac{1}{2}}}{N_1|\Sigma_1|^{\frac{1}{2}}} \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \tau \tag{23}$$

For the case of using the Mahalanobis distance, given by: $d_i^2(Y,X) = (Y-X)^T\Sigma_i^{-1}(Y-X)$, the volume $\nu_i(X)$ is given by: $\nu_i = \pi^{n/2}\Gamma^{-1}(n/2+1)|\Sigma_i|^{1/2}d_i^n$ (where n is the number of features) [6]. If the a priori class probabilities are equal, the decision is made according to the minimal volume of the class.

Multilayer perceptron. Neural networks offer a valuable alternative to Bayesian classifiers in evaluating a posteriori class probabilities for classifying stochastic patterns. In contrast to the Bayesian parametric classifier,

the "neural" classifier makes no assumptions on the probabilistic nature of the problem, and is thus universal in the sense that it is not restricted to an underlying probabilistic model. Instead, it adjusts itself to a given training data set by a training algorithm, and thus, can learn the stochastic properties of the specific problem. Multilayer perceptron (MLP), probably the best known type of neural nets, with back–propagation training algorithm [13] is used in this paper as a "neural" nonparametric pattern classifier.

The typical back-propagation network always has an input layer, an output layer and at least one hidden layer (typically one or two). Each layer is fully connected to the succeeding layer. During training, information is propagated back through the network and used to update the connection weights. For the purpose of brief describing the basis back-propagation algorithm, we shall accept a notation as follows: $x_j^{[s]}$- current output state of jth neuron in layer s, $w_{ji}^{[s]}$-weight on connection joining ith neuron in layer $(s-1)$ to jth neuron in layer s, and $I_j^{[s]}$- weighted summation of inputs to jth neuron in layer s. A back-propagation element therefore transfers its inputs as follows

$$x_j^{[s]} = f(I_j^{[s]}) = f(\sum_i (w_{ji}^{[s]} x_i^{[s-1]})) \tag{24}$$

where f is traditionally the sigmoid function but can be any differentiable function, see Fig. 1. The sigmoid function is defined as: $f(z) = (1+e^{-z})^{-1}$.

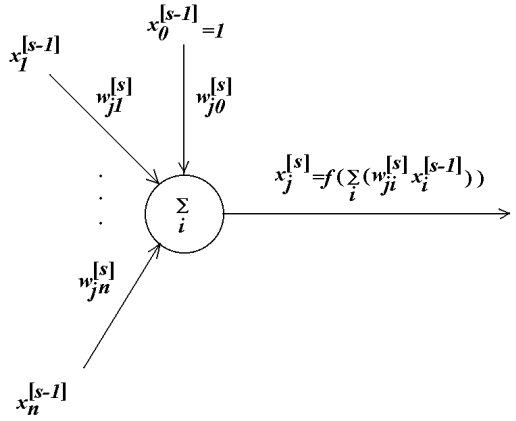

Fig. 1. A typical back-propagation processing element

Suppose now that the network has some global error function E associated with it which is a differentiable function of all the connection weights in

the network. The critical parameter that is passed back through the layers and considered as a measure of the local error at processing element j in layer s is defined by

$$e_j^{[s]} = -\frac{\partial E}{\partial I_j^{[s]}} = f'(I_j^{[s]}) \sum_k (e_k^{[s+1]} w_{kj}^{[s+1]}) \tag{25}$$

Note that in (25), there is a layer above layer s; therefore, (25) can only be used for non-output layers. If f is the sigmoid function, then its derivative can be expressed as a simple function of itself as follows: $f'(z) = f(z)(1 - f(z))$, which leads to the following form of (25)

$$e_j^{[s]} = x_j^{[s]}(1 - x_j^{[s]}) \sum_k (e_k^{[s+1]} w_{kj}^{[s+1]}) \tag{26}$$

The summation term in (26) which is used to back-propagate errors is analogous to the summation term in (24) which is used to forward propagate the input through the layers to the output layer, determine the error at the output layer, and then propagate the errors back through the network from the output layer to the input layer using (26) or more generally (25). The aim of the training process is to minimize the global error E of the system by modifying the weights. Given the current set of weights $w_{ji}^{[s]}$, we need to determine how to increment or decrement them in order to decrease the global error. This can be done by using a gradient descent rule as follows

$$\Delta w_{ji}^{[s]} = -lcoef \frac{\partial E}{\partial w_{ji}^{[s]}} = lcoef \cdot e_j^{[s]} x_i^{[s-1]} \tag{27}$$

where $lcoef$ is a "learning rate" coeficient. Suppose a vector i is presented at the input layer and suppose a teacher specifies the desired output d. Let o denote the actual output produced by the network with its current set of weights. Then a measure of the error in achieving that desired output is given by: $E = 0.5\Sigma((d_k - o_k)^2)$, where the subscript k indexes the components of d and o. From (25), the scaled "local error" at each processing element of the output layer is given by

$$e_k^{[o]} = -\frac{\partial E}{\partial I_k^{[o]}} = -\frac{\partial E}{\partial o_k}\frac{\partial o_k}{\partial I_k} = (d_k - o_k) f'(I_k) \tag{28}$$

As a summary, for given an input vector i and a desired output vector d, we should do the following:

1. Present i to the input layer and propagate it through to the output layer to obtain an output vector o.
2. As this information propagates through the network, it will also set all the summed inputs I_j and output states x_j for each processing element in the network.
3. For each processing element in the output layer, calculate the scaled local error as given (28) and then calculate the delta weight using (27).
4. For each layer s, starting at the layer below the output layer and ending with the layer above the input layer, and for each processing element in layer s, calculate the scaled local error as given in (26), then calculate the delta weight using (27).
5. Update all weights in the network by adding the delta weights to the corresponding previous weights.

5. Experimental analysis

The experimental part of the work is based on photographs of $3D$ objects, Fig. 2, belonging to five classes picked up from real scenes with VHS TV camera. Analog signals are digitized by a frame grabber PC board and Imaging Technology 151 processing system. Forty-seven photos are picked up for every object class [12]. The nonoccluded objects are rotated by arbitrary angles, shifted within the image plane and scaled by camera shift. The resulting gray level images are thresholded to produce binary silhouettes (see Fig. 3 for an example). A boundary follower algorithm [12] is used to identify the boundary contours, the elements of which are used for the feature extraction (see Fig. 4).

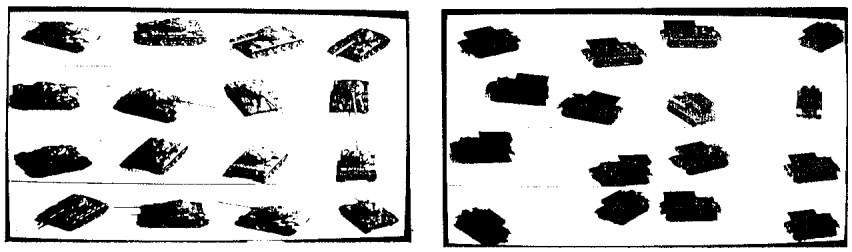

Fig. 2. 3D object photos used in the experiments

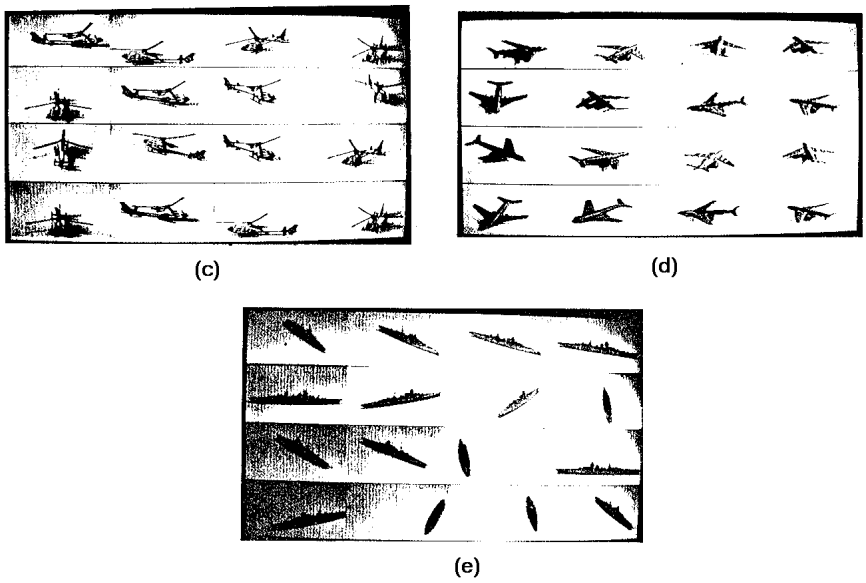

Fig. 2. Continue

In this paper, the performance of the Bayesian parametric ("piecewise" and "pairwise" linear, as well as quadratic classifier) and nonparametric classifiers ($k-NN$ "voting" and "volumetric" classifiers, as well as MLP) are analyzed. All classification experiments were performed by using a specialized interactive PC software package for pattern recognition, PC-PARIS [11]. The performance of the $k-NN$ classifiers is considered for three values of the k nearest neighbors: 3, 5, and 9. As for MLP, in all of the experiments we fixed, on the basis of an extensive preliminary experimental analysis, the following parameters: $lcoef = 0.01$, initial weights $[w_0] = [0.5]$, and one or two hidden layers with 10 nodes. As references for the quality analysis of the applied classifiers, we have used the corresponding Bayes error estimates obtained by a very accurate $k-NN$ procedure combining the resubstitution and leave-one-out methods [6]. The final Bayes error estimates are obtained by averaging the obtained $k-NN$ error estimates for the values of k ranging from 3 to 43.

In Table 1, the percentual values of the classification errors for the proposed classifiers obtained in classifying data for both 12-dimensional feature vectors of the moment invariants and 11−dimensional ones of the Zernike moment invariants are presented. Based on the results presented in Table 1,

it can be concluded that best results are obtained by using the MLP with two hidden layers (MLP(2)) and the moment invariants as feature extraction method.

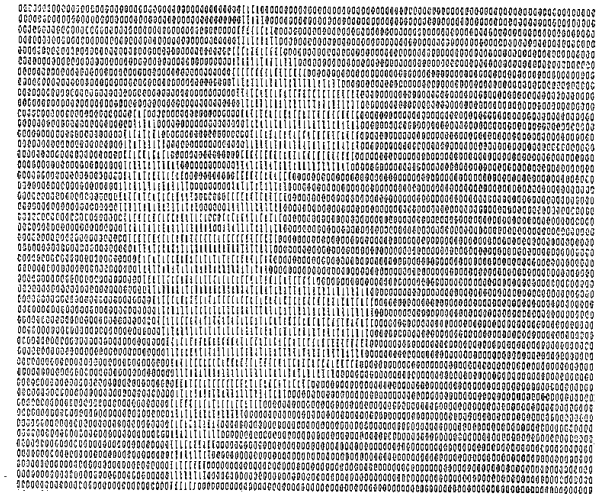

Fig. 3. An example of the object silhouette

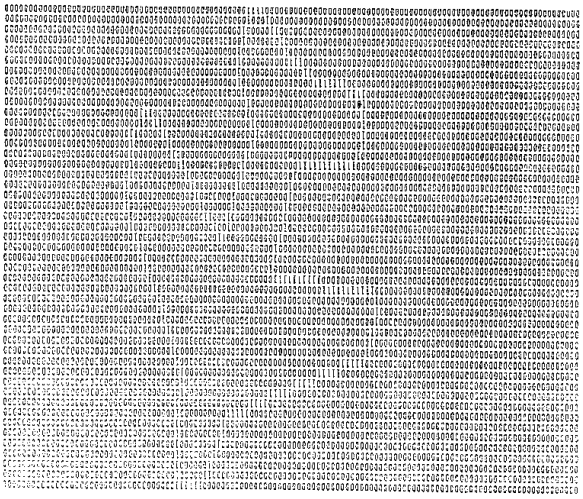

Fig. 4. Object boundary obtained by using the original boundary follower algorithm

As for the AR model feature extraction methods, two groups of feature vectors, obtained by using 1D and 2D AR models, are used. The estimated

parameters of these models represent the feature vectors for classification. The number of features is in the range from 1 to 18.

Table 1: Classification errors (in percents) obtained by the proposed classifiers: Moments.

Classifiers		Moment invariants	Zernike Moments
Bayes error estimate		1.84	2.38
Linear classifier (piecewise)		12.77	10.64
Linear classifier (pairwise)		7.66	8.94
Quadratic classifier		3.83	3.40
$k-NN$ voting	3	5.53	14.47
	5	6.38	16.60
	9	8.51	20.00
$k-NN$ volumetric	3	5.53	18.30
	5	6.38	25.11
	9	6.81	30.64
MLP hiden layers	1	5.11	26.38
	2	0.85	21.70

In Table 2, the percentual values of the classification errors obtained by using both AR models are presented. It can be concluded that the best results are obtained by using $3-NN$ volumetric classifier and the 2D AR model of the first order. As for AR models with higher orders, the best classification results are obtained by using the MLP(2) in the case of 2D AR model of the 8th order. For lower orders of the 2D AR model and for 1D AR model, we could not obtain the satisfactory results by using the MLP.

Table 2. Classification errors (in percents) obtained by the proposed classifiers: 1D and 2D AR model.

		Feature numb. - 1D AR model						Feature numb. - 2D AR model							
Classifiers		1	2	3	4	5	6	4	6	8	10	12	14	16	18
B. E.		13.7	9.6	33.0	22.1	15.3	6.3	2.8	7.9	12.7	14.4	13.5	9.5	7.6	6.0
L. C. (piec.)		36.2	32.3	35.7	34.9	34.9	25.5	30.6	37.9	29.9	31.9	28.5	24.7	25.5	25.1
L. C. (pair.)		30.2	21.7	34.5	30.2	25.5	12.3	27.7	36.2	23.8	21.7	17.0	17.9	17.0	13.6
Q. C.		26.4	20.0	37.0	28.1	23.4	17.9	39.6	30.6	20.8	13.6	14.9	11.9	7.7	7.7
k-NN vot.	3	12.8	12.8	28.5	31.5	39.6	42.1	2.1	22.1	22.1	22.1	23.0	19.6	14.5	18.0
	5	14.0	16.6	36.6	40.0	46.4	48.9	2.5	24.7	23.8	26.0	24.7	20.4	17.4	19.6
	9	21.7	26.8	43.0	45.5	51.9	51.5	5.1	28.5	24.7	27.2	25.5	22.5	14.5	17.0
k-NN vol.	3	11.5	8.5	29.4	20.0	17.9	17.9	1.3	12.8	16.6	19.6	17.9	14.5	11.9	10.6
	5	12.3	11.9	33.6	26.4	24.3	25.5	1.7	14.0	21.7	23.0	19.6	17.4	14.5	11.5
	9	13.2	15.3	34.5	29.8	28.5	28.1	3.8	17.4	26.4	25.1	21.7	17.9	14.5	11.5
MLP h.l.	1	68.5	58.7	40.4	40.0	37.4	23.0	64.3	64.3	34.5	43.1	23.2	34.5	30.2	5.5
	2	60.0	61.3	55.7	60.8	43.0	19.2	66.4	66.4	40.4	23.8	15.9	35.3	33.2	1.7

Based on the entire experimental analysis, it could be concluded that the most appropriate feature extraction/classification combination for solving the 2D contour classification example considered in this paper, consists of the moment invariants and MLP with 2 hidden layers. In this sense, this combination is recommended for solving 2D contour classification problems where highly limited training data sets are available.

6. Conclusion

In this paper, the application of statistical parametric and nonparametric pattern classifiers to the classification of 2D shapes represented by the moment invariants and the stochastic AR models is considered. Performance of the considered classifiers is analyzed with respect to the corresponding Bayes error estimates, calculated for the given data set. Based on the entire experimental analysis, the moment invariants, as the feature extraction method, and the multilayer perceptron with 2 hidden layers, as the pattern classifier, are recommended for 2D shape classification problems in highly limited training data set conditions.

REFERENCES

1. S. Belkasim, M. Shridhar and M. Ahmadi: *Pattern Recognition with Moment Invariants: A Comparative Study and New Results.* Pattern Recognition, Vol. 24, No. 12, pp. 1117-1138, 1991.

2. M. Das, M.J. Paulik and N.K.Loh: *A Bivariate Autoregressive Modeling Technique for Analysis and Classification of Planar Shapes.* IEEE Trans. on PAMI, Vol. 12, No. 1, January 1990.

3. S.R. Dubois and F.H. Glantz: *An Autoregressive Model Approach to Two-Dimensional Shape Classification.* IEEE Trans. on PAMI, Vol. PAMI-8, No.1, January 1986.

4. R.O. Duda and P.E. Hart: *Pattern Classification and Scene Analysis.* New York, Wiley, 1973.

5. S. Dudani, K.Breeding, R.McGhee: *Aircraft Identification by Moment Invariants, IEEE Trans. on Computers.* Vol. C-26, 1977, pp. 39-45.

6. K. Fukunaga: *Introduction to Statistical Pattern Recognition.* ACADEMIC PRESS INC., Harcourt Brace Jovanovich, Publishers, Boston - San Diego - New York - London - Sydney - Tokyo - Toronto, 1990.

7. M.K. Hu: *Visual Patern Recognition By Moment Invariants.* IEEE Trans. on Information Theory, Vol. IT-8, Feb. 1962, pp. 179-187.

8. R.L. Kashyap and R. Chellappa: *Stochastic Models for Closed Boundary Analysis: Representation and Reconstruction.* IEEE Trans. on Information Theory, Vol. IT-27, No. 5, September 1981.

9. A. Khotanzad and Y.Poggios: *Invariant Image Recognition by Zernike Moments.* IEEE Trans. on PAMI, Vol. PAMI-12, 1990, pp. 489-497.

10. L. Ljung: *System Identification: Theory for the User.* Englewood Cliffs, NJ: Prentice-Hall, 1987.

11. M. Milosavljević and M. Marković: *A PC-Based Software Tool for Statistical Pattern Classifier Design.* in Proc. of the Third IEEE International Conference on Electronics, Circuits, and Systems, ICECS'96, October 13-16, Rodos, Greece, Vol. 2, pp. 720-723.

12. P. Pejnović, Lj. Buturović and Z. Stojiljković: *Object Recognition by Invariants.* in Proc. of 11th IAPR Inter. Conference on Pattern Recognition, Vol. II, pp. 434-437, 1992.

13. D.E. Rumelhart, G.E. Hinton and R.J. Williams: *Learning internal representations by error propagation.* In D.E.Rumelhart and J.L.McClelland (eds), Parallel Distributed Processing: Exploration in the Microstructure of Cognition, Vol. 1, Chap. 8, MIT Press/Bradford Books. 1986.

14. I. Sekita, T. Kurita and N. Otsu: *Complex Autoregressive Model for Shape Recognition.* IEEE Trans. on PAMI, Vol. 14, No. 4, April 1992, pp. 489-496.

15. V.A. Spasić: *Problems in mathematical modeling of live systems.* Journal of Medical & Biological engineering & Computing, 1977, Vol. 35, Supp. 1, pp. F.70-051.04, ISSN 0140-018/97.

16. M.R. Teaque: *Image analysis via the general theory of moments.* J. Opt. Soc. Am., Vol. 70, No. 8, August 1980.

17. Q.M. Tieng and W.W. Boles: *Recognition of 2D Object Contours Using the Wavelet Transform Zero-Crossing Representation.* IEEE Trans. on PAMI, Vol. 19, No. 8, 1997, pp. 910-916.

18. C.T. Zahn and R.Z. Roskies: *Fourier descriptors for plane closedcurves.* IEEE Tran. On Computers, Vol. 21, No. 3, 1972, pp. 269-281.