

# Fusion of probabilistic knowledge-based classification rules and learning automata for automatic recognition of digital images

Darío Maravall    Javier de Lope    Juan Pablo Fuentes

## A B S T R A C T

In this paper, the fusion of probabilistic knowledge-based classification rules and learning automata theory is proposed and as a result we present a set of probabilistic classification rules with self-learning capability. The probabilities of the classification rules change dynamically guided by a supervised reinforcement process aimed at obtaining an optimum classification accuracy. This novel classifier is applied to the automatic recognition of digital images corresponding to visual landmarks for the autonomous navigation of an unmanned aerial vehicle (UAV) developed by the authors. The classification accuracy of the proposed classifier and its comparison with well-established pattern recognition methods is finally reported.

## 1. Probabilistic Knowledge-based Classification Rules with Self Learning Capability

Knowledge-based classification rules of the type *If {Condition} Then {Class}* have been widely used in Machine Learning and Pattern Recognition problems. In this paper we propose to enlarge knowledge-based classification rules with associated probabilities that can dynamically change in a supervised fashion in order to improve the classification accuracy of the global set of rules.

We define a stochastic set of knowledge-based classification rules as a conventional base of knowledge rules in which each rule  $R_i$  has an associated probability  $p_i(k)$ . These probabilities change at each iterative instant  $k$  as a consequence of a supervised reinforcement learning process to be explained later on. Our proposed classification rules have the following conjunctive normal form (CNF) generic structure:

$$R_i : \text{If } x_1 \text{ is } A \wedge x_2 \text{ is } B \wedge \dots \wedge x_n \text{ is } Z \text{ Then Class } C_s; p_i(k) \quad (1)$$

where  $x_1, x_2, \dots, x_n$  are the discriminant variables or features,  $A, B, \dots, Z$  are either fuzzy or Boolean subsets in which the features have been partitioned (interval or boxes-like partitions) and  $p_i(k)$  is the probability associated to rule  $R_i$  at the iterative generic instant  $k$ . These probabilities are a key element in the proposed classifier as they determine the strength or validity of the corresponding rule and change dynamically by virtue of a supervised reinforcement learning process aimed at optimizing the classifier accuracy.

It can be noticed that thanks to the above mentioned conjunctive normal form structure of the antecedent clauses of the classification rules these antecedent clauses can be straightforwardly equated to basic, atomic information granules according to the well-known granular computing paradigm [1].

Due to its fundamental role in the proposed classifier, in the next paragraph we explain in detail the process of building the antecedent clauses of the classification rules.

## 2. Granulation process to build the antecedent clauses of the classification rules

The granulation process is a fundamental phase in any particular classification problem. Since each discriminant variable or feature is usually represented by numeric values, we must create a set of clusters for each variable. If we define a large number of clusters, we will increase the detail in the feature description. However, it also affects to the size and performance of the classifier. Thus, we must take into account this important trade off at design stage. Basically, the granulation process is formed by the following steps:

- Step 1 Selection of the most representative or principal state variables of the system  $x_1, x_2, \dots, x_n$ , which are the  $n$  principal state variables that optimize the representation of all the indistinguishable states or clusters. Obviously, in the particular case of pattern recognition systems, the state variables are usually known as discriminant variables.
- Step 2 Granulation or partition of all the state variables. In most cases before we have usually applied a Boxes-like [2] based granulation of the state variables with satisfactory

results and in the application considered in this paper we have also applied the same idea as explained below. Roughly speaking, the interval granulation is a hard open problem in every particular application, which can be solved either by hand by the designers themselves if they have the necessary information or by means of an automated off-line optimization process based on e.g. evolutionary techniques. In this paper we have applied a mixture of both approaches.

Concretely we divide each discriminant variable  $x_i$  at hand into the three following granules or intervals: *low* ( $x_i < \bar{x}_i - k\sigma_i$ ), *middle* ( $\bar{x}_i - k\sigma_i \leq x_i \leq \bar{x}_i + k\sigma_i$ ), *high* ( $\bar{x}_i + k\sigma_i < x_i$ ), as displayed in Fig. 1, where  $\bar{x}_i$  is the mean value of the discriminant variable  $x_i$ ,  $\sigma_i$  is its standard deviation, and  $k_i$  is a real parameter to be optimized. This parameter is defined as *granulation constant*.

Once the principal state variables have been selected and partitioned in the corresponding intervals then the basic information granules or clusters emerge as a consequence of the processing of the labeled exemplars taken from the training dataset, so that the next step is to map the cluster space  $\mathcal{X} = x_1, x_2, \dots, x_n$  to the classes space  $\mathcal{C} = C_1, C_2, \dots, C_r$ .

In the sequel we describe the basics of the method that we propose for this mapping process, which is simply based on the idea of modeling the information granules embedded in the antecedent clauses of the classification rules as the states of a learning automaton operating in a random environment, so that the classification rules probabilities are updated by means of a conventional automata's reinforcement learning algorithm as explained below (notice that in this case rewards and penalties from the environment are equivalent to correct and incorrect classification, respectively).

### 3. A brief description and introduction to learning automata theory

Learning automata, also called stochastic automata with variable structure, have received considerable attention as models of adaptive and learning systems [3].

#### 3.1. Basic definitions

A learning automaton is a sextuplet  $\langle x, Q, u, \bar{P}(t), G, \mathcal{R} \rangle$ , where  $x$  is the finite set of inputs,  $Q = \{q_1, q_2, \dots, q_m\}$  is a finite set of internal states,  $u$  is the set of outputs,  $\bar{P}(t) = \|p_1(t), p_2(t), \dots, p_m(t)\|$  is the state probability vector at time instant  $t$ ,  $G: Q \rightarrow u$  is the output function (normally considered as deterministic and one-one), and  $\mathcal{R}$  is an algorithm called the reinforcement scheme, which generates  $\bar{P}(t+1)$  from  $\bar{P}(t)$  and the particular input at a discrete instant  $t$ .

The automaton operates in a random environment and chooses its current state according to the input received from the environment. The new state probabilities distribution  $\bar{P}(t+1)$  reflects the information obtained from the environment. The random environment has a set of inputs  $u$  and its set of outputs is frequently binary  $\{0, 1\}$ , with '0' corresponding to the reward response and '1' to the penalty response. If the input to the environment is  $u_i$  the environment produces a penalty response with probability  $c_i$ .

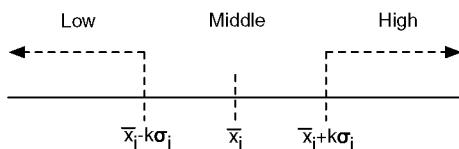


Fig. 1. Granules or intervals in which the discriminant variables have been divided.

Fig. 2 shows the feedback configuration of a learning automaton operating in a random environment. At each instant  $t$  the environment evaluates the action of the automaton by either a penalty '1' or a reward '0'. The performance of the automaton's behavior is the average penalty

$$I(t) = \frac{1}{m} \sum_{i=1}^m p_i(t) c_i \quad (2)$$

which must be minimized. In order to minimize the expectation of penalty (Eq. 2), the reinforcement scheme modifies the state probability vector  $\bar{P}(t)$ . The basic idea is to increase  $p_i$  if state  $q_i$  generates a reward and to decrease  $p_i$  when the same state has produced a penalty. A great number of reinforcement schemes for minimizing the expected value of penalty have been studied and compared. One of the most serious difficulties that arise in learning automata is the dichotomy between learning speed and accuracy. If the speed of convergence is increased in any particular reinforcement scheme, this action is almost invariably accompanied by an increase of convergence to the undesired state [4,5].

In the next paragraph 3.2 we present the properties of a novel reinforcement scheme, previously introduced by ourselves in [6], that makes it possible to control simultaneously speed and accuracy of learning.

#### 3.2. Stochastic reinforcement algorithms in learning automata theory

In the technical literature a widely used stochastic reinforcement algorithms is  $L_{R-I}$ , which stands for Linear Reward-Inaction algorithm.

Let us suppose that the action chosen by the automaton at instant  $k$  is  $\phi_i$ , for the  $L_{R-I}$  the updating of the action probabilities is as follows:

$$p_i(k+1) = p_i(k) + \lambda \beta(k) [1 - p_i(k)] \quad (3)$$

$$p_j(k+1) = p_j(k) - \lambda \beta(k) p_j(k) \quad \forall j \neq i, 1 \leq j \leq N \quad (4)$$

where  $0 < \lambda < 1$  is the learning ratio and  $\beta(k)$  is the environment's response:  $\beta = 1$  (favorable response or reward) and  $\beta = 0$  (unfavorable response or penalty in which case the algorithm do not change the probability, i.e. inaction).

#### 3.3. A novel R-L algorithm: the incremental R-L Algorithm

We have previously proposed a novel reinforcement algorithm for learning automata: the so-called Incremental R-L algorithm [6] that we describe in the sequel.

This algorithm is based on the dynamic properties of the following discrete stochastic process:

$$p_i(k+1) = p_i(k) + p_i(k) \sum_{i=1, j \neq i}^m a_{ij} p_j(t) \quad 1 \leq i \leq m \quad (5)$$

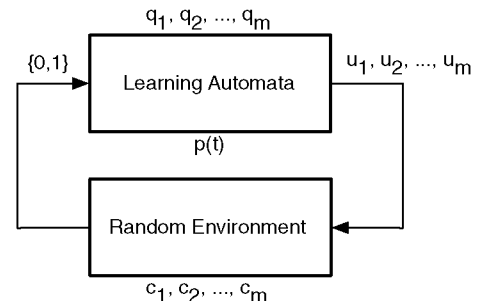


Fig. 2. Interaction of learning automaton with random environment.

which is a set of  $m$  nonlinear difference equations between probabilities. Here we are only concerned with the most interesting properties of (5) from the standpoint of its application to learning automata. These properties are presented in the sequel.

### 3.3.1. Constraints on the incremental R-L algorithm parameters

The necessary and sufficient conditions that guarantee the probabilistic existence field of (5) are given by

$$a_{ij} = -a_{ji} \quad \text{and} \quad \max |a_{ij}| < 1 \quad \text{for } 1 \leq i, j \leq m \quad (6)$$

We have proved [7] that if (6) holds then

$$\forall i: 0 \leq p_i(t) \leq 1 \quad \text{and} \quad \sum_{i=1}^m p_i(t) = 1 \quad (7)$$

### 3.3.2. Solutions of stochastic process

The system of difference Eq. (5) has the following  $m$  extreme solutions:

$$(1, 0, \dots, 0); (0, 1, \dots, 0); \dots (0, 0, \dots, 1) \quad (8)$$

If  $m$  is odd then there exists an additional intermediate solution which may or may not belong to the existence field of (5). When  $m$  is even the existence of the additional solution is not always assured. A remarkable result derived from the above constraints on the  $a_{ij}$  parameters given by expression (9) is the additional intermediate solution is unstable on nonabsorbing (see paragraphs 3.3.3 and 3.3.4 below).

### 3.3.3. Global convergence of stochastic process

This property is of considerable importance in order to apply (5) as a reinforcement scheme of learning automata. It can be proved that, forming the determinant of (5)

$$\begin{vmatrix} 0 & a_{12} & a_{13} & \dots & a_{1m} \\ -a_{12} & 0 & -a_{23} & \dots & a_{2m} \\ \vdots & \vdots & \vdots & & \vdots \\ -a_{1m} & -a_{2m} & -a_{3m} & \dots & 0 \end{vmatrix} \rightarrow \begin{matrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{matrix} \quad (9)$$

if one row of (9), and only one is each case, has all its elements positive, then the system of (5) converges globally to the respective solution. That is, for fixing ideas, if all coefficients in the first row  $a_{12}, a_{13}, \dots, a_{1m}$  are positive, then the stochastic process converges unconditionally to the extreme position  $(1, 0, \dots, 0)$ , whatever the system's starting position would be.

### 3.3.4. Nonabsorbing nature of stochastic process

According to the above property the  $a_{ij}$  coefficients govern the convergence of the stochastic process of (5) to one of its  $m$  internal states. Furthermore, the extreme solutions are absorbing. This fact is an undesirable characteristic from the learning automaton (whose state probabilities will hold the set of equations given by Eq. 5) when it locks on a particular state. If the random environment's penalty likelihoods change, and so does the optimum state, absorption prevents the automaton tending to the new optimum state. Nevertheless, it can be proved that the system (5) does not reach any extreme solution in a finite time.

### 3.3.5. Incremental reinforcement scheme

The unconditional convergence of the stochastic process of (5) to one of the extreme solutions is vital in order to establish an algorithm for the updating of the state probabilities. Since it is desired to obtain a convergence to the less punished state with probability 1, one can plan to adapt the state probabilities of the automaton to the system (5). Furthermore, as the coefficients  $a_{ij}$  control the convergence the basic idea behind the new algorithm is rather simple.

If the automaton selects a state  $q_k$  at instant  $t$  and a reward occurs, the coefficients  $a_{kj}$ , with  $1 \leq j \leq m$ , are increased. When the same state is chosen but a penalty occurs then the proper action is to decrease the  $a_{kj}$ . Therefore, if the automaton has selected the state  $q_i$  at a generic instant  $t$  and a reward has occurred, the state probabilities are given by

$$\begin{aligned} p_i(t+1) &= p_i(t) + \sum_{j=1, j \neq i}^m (a_{ij} + \Delta a_{ij}) p_i(t) p_j(t) p_h(t+1) \\ &= p_h(t) + \sum_{j=1, j \neq i, h}^m a_{hj} p_h(t) p_j(t) \\ &\quad + (a_{hi} - \Delta a_{hi}) p_h(t) p_i(t) \quad 1 \\ &\leq h \leq m; h \neq i \end{aligned} \quad (10)$$

when the same state produces a penalty from the environment, the above expressions are valid, except that the sign of the increments  $\Delta a_{ij}$  and  $\Delta a_{ih}$  must be changed.

### 3.3.6. Comparison between incremental and classical reinforcement schemes

We shall only comment on the most significant experimental results obtained from computer simulations performed in order to compare the incremental algorithm with the reward-inaction scheme, which was chosen because of its excellent properties [3]. The parameters taken into account were (a) level of learning, and (b) the average penalty received by the automaton. The results can briefly be summarized as follows: (i) As far as the number of trials is concerned, the automaton's behavior is slightly better when the incremental scheme is applied. (ii) The minimization of the average penalty is very similar for the two algorithms. (iii) For a nonstationary environment the incremental scheme produces a much better performance of the automaton. In our opinion this is the main reason that justify the use of the new algorithm.

## 4. Automatic recognition of visual landmarks in topological maps

We have applied the classifier proposed in this paper to the recognition of visual landmarks of the topological map used for the autonomous navigation of an unmanned aerial vehicle (UAV). The UAV is a conventional AR. Drone by Parrot [8] depicted in Fig. 3.

We have presented elsewhere [9,10] an approach based on evolutionary reinforcement learning for the autonomous navigation of an UAV. However, in this paper the UAV's navigation system is based on a visual topological map implemented as a labeled graph in which the nodes correspond to visual landmarks and the edges or arcs to specific UAV's control maneuvers.

The UAV's navigation system utilizes the front on-board camera, which is a HD camera that is able to capture  $640 \times 360$  images at 30 fps. The images are used by the UAV's controller, which is based on a dual feedforward/feedback architecture [11,12], to generate the control commands in real-time. The system processes the images and generates new control signals each 30 ms. Fig. 4 displays the dual feedforward/feedback controller in which the error signal is obtained as the histogram of the goal landmark image minus the histogram of the current image captured by the UAV.

The visual error signal and the corresponding control actions during the UAV's maneuver of *door approximation and crossing* are displayed in Fig. 5. This maneuver is one of the basic behaviors in UAV [13] as well as in conventional autonomous robots [14]. In Fig. 6 a sequence of the images captured by the UAV's on-board camera while performing the *door approximation and crossing maneuver* is also shown.





Fig. 3. The Parrot AR.Drone 2.0 prior to takeoff in one of the experiments.

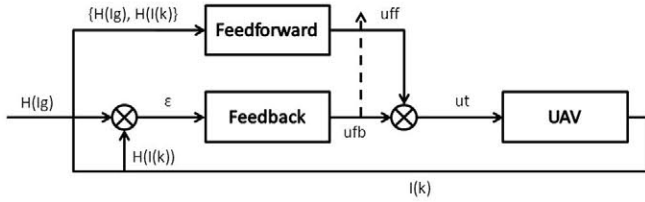


Fig. 4. The feedforward/feedback controller. The error signal of both controllers is obtained as the difference between the histogram of the recognized landmark, i.e. the goal image,  $H(lg)$  and the histogram of the current image  $H(l(k))$ .

Obviously, for an efficient control of the UAV it is vital a correct recognition of the visual landmarks to generate the proper signal error to the UAV's controller. In the sequel we describe our experiments on visual landmarks recognition in which we have compared the recognition accuracy of the classifier proposed in this paper with another well-known pattern recognition techniques.

First, we have defined four different landmarks on the UAV's topological map: a door, a decorating figure, a TV set and a picture. Fig. 7 shows some images of these four landmarks. For landmarks recognition we have applied as discriminant variables several descriptors of the images histogram: standard deviation, skewness, kurtosis, uniformity and entropy.

#### 4.1. Discriminant features applied for landmark recognition

For the recognition of the landmarks we have applied as discriminant variables several descriptors of the images histogram  $h(k)$ : standard deviation  $\sigma$ , skewness  $S$  (13), kurtosis  $K$  (14), uniformity  $U$  (15) and entropy  $E$  (16), where  $p(k)$  is the normalized histogram (11),  $\mu_n$  is the statistical moment of the normalized histogram (12) and  $\sigma^2 = \mu_2$ .

$$p(k) = \frac{h(k)}{\sum_{i=1}^N h(i)} \quad (11)$$

$$\mu_n = \frac{1}{N} \sum_{k=1}^N (k - m)^n p(k) \quad (12)$$

$$S = \frac{\mu_3}{\sigma^3} \quad (13)$$

$$K = \frac{\mu_4}{\sigma^4} \quad (14)$$

$$U = \sum_{k=1}^N p^2(k) \quad (15)$$

$$E = -\sum_{k=1}^N p(k) \log_2 p(k) \quad (16)$$

In the next section we report and comment the experimental results that we have obtained with the classifier proposed in this paper for the recognition of the above mentioned visual landmarks. We also compare these results with those obtained with other standard pattern recognition methods like the minimum Euclidean distance and the  $k$ -nearest neighbor algorithm.

### 5. Experimental results

Each experiment consists of building a classifier and getting its averaged true classification ratio (which is the complementary of the true classification error) with the dataset described above (see Fig. 7). We employ the leave-one-out cross validation technique. Each experiment is run 200 times. We report the average and peak results.

The proposed classifier gets an averaged true classification ratio of 0.813 and a peak ratio of 0.875. The best performance is obtained with a granulation constant  $k = 0.4$  and a learning ratio

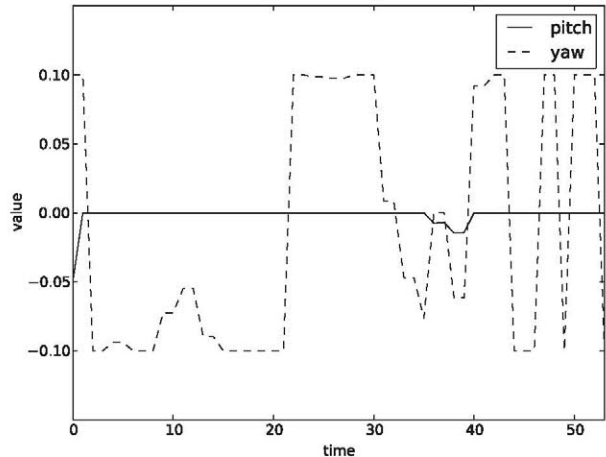
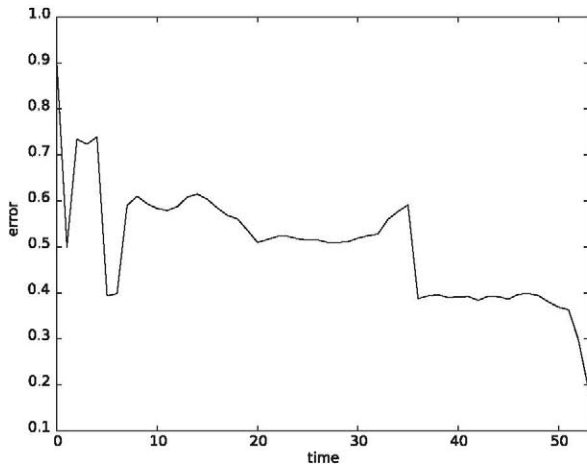
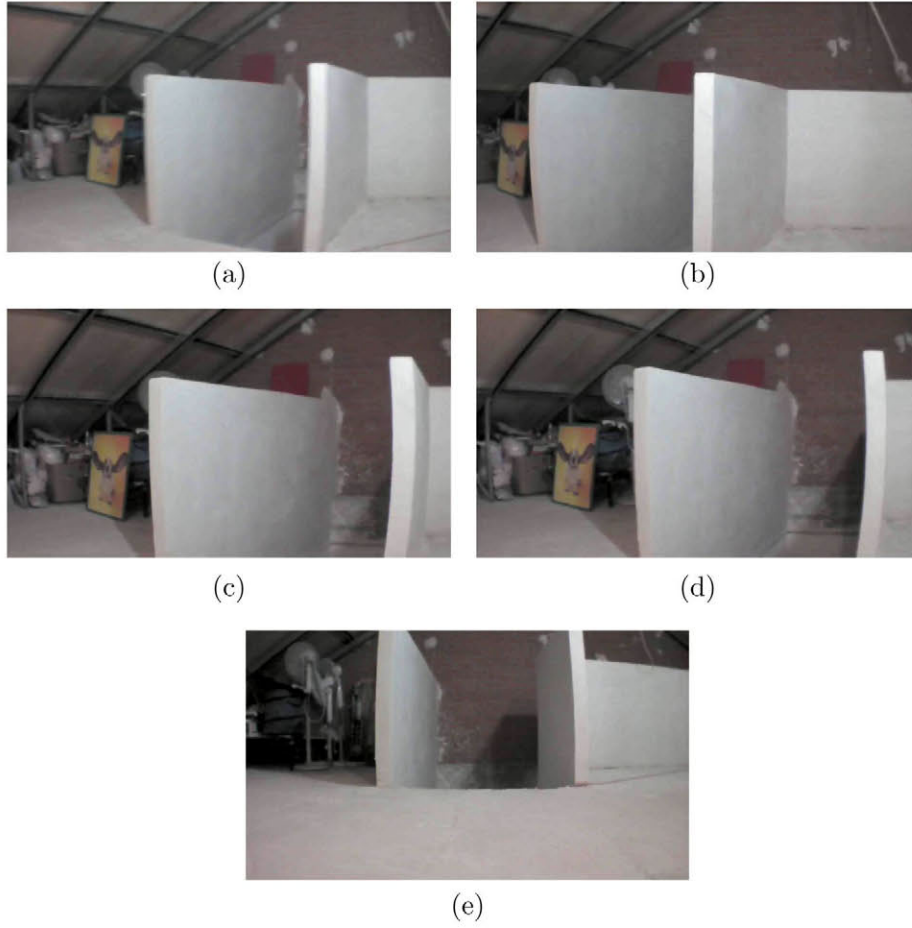
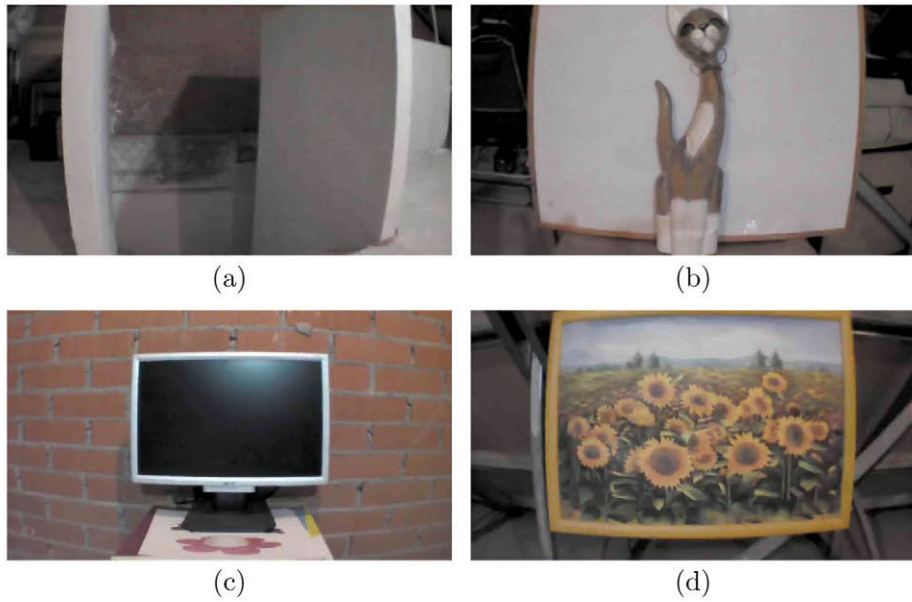


Fig. 5. The visual error signal during the door-landmark approximation and crossing maneuver (left) and the control signals during the door approximation and crossing maneuver (right).



**Fig. 6.** The sequence includes the successive images captured by the UAV while performing a door approximation and crossing maneuver.



**Fig. 7.** Some examples of the landmarks images of the dataset used for the design and testing of the classifiers.

$\lambda = 0.25$ . All these parameters have been optimized by means of an evolutionary algorithm.

By analyzing the confusion matrix shown in Table 1 we can observe that the greatest number of errors are produced by the fourth

landmark in the dataset shown in Fig. 7 (i.e. the picture), which is erroneously classified 40% that is recognized as the exit door (i.e. first landmark). This confusion matrix corresponds to the best classifier. After analyzing the results and the description vectors,

**Table 1**  
Probabilistic knowledge-based classifier confusion matrix.

	Landmark 1	Landmark 2	Landmark 3	Landmark 4
Landmark 1	8	0	1	1
Landmark 2	0	10	0	0
Landmark 3	1	0	9	0
Landmark 4	4	0	0	6

**Table 2**  
True classification ratios of the probabilistic knowledge-based classifier and other traditional methods.

Classifier	Averaged ratio
Probabilistic knowledge-based	0.813
Minimum distance	0.475
$k$ -Nearest neighbor ( $k = 3$ )	0.700
$k$ -Nearest neighbor ( $k = 5$ )	0.675
$k$ -Nearest neighbor ( $k = 7$ )	0.600

probably it is not an issue in the classifier and it could be produced by the descriptors that we have selected.

We have also computed the results achieved by conventional, well-established pattern recognition methods as the minimum distance classifier and  $k$ -nearest neighbor algorithm. We have used Euclidean distances in all the traditional methods. The results are summarized in Table 2.

## 6. Conclusions and further work

A novel pattern recognition method based on the fusion of probabilistic knowledge-based classification rules and learning automata theory has been introduced and tested with a challenging image recognition problem, in which we have used several statistical histogram-based descriptors, namely standard deviation, kurtosis, skewness, uniformity and entropy, as discriminant features. We think that this idea of using such a reduced number of discriminant variables for the recognition of complex images is

of great practical interest for other computer vision applications. Furthermore, as this reduced set of discriminant variables are based on the image histogram, makes them even more attractive due to the well-known robustness of the image histogram concerning illumination and perspective changes as compared to the image itself. Due to the excellent results obtained with this reduced set of discriminant variables and with the proposed classifier for our images data set we plan to extend in the future its application and testing to other publicly available standard image data sets.

## References

- Bargiela, A., Pedrycz, W., J. Granular Computing. An Introduction. Kluwer Academic Publishers., Boston.
- Michie, D., Chambers, R.A., J. Boxes: an experiment in adaptive control. In: Dale, E., Michie, D. (Eds.), Machine Intelligence 2. Oliver & Boyd, pp. 137–152.
- Narendra, K., Thathachar, M., J. Learning automata – A survey. IEEE Transactions Systems Man and Cybernetics 4 (4), 323–334.
- Narendra, K., Wright, E., Mason, L., J. Applications of learning automata to telephone traffic routing and control. IEEE Transactions on Systems Man and Cybernetics 7 (11), 785–792.
- Narendra, K., Viswanathan, R., J. A two-level system of stochastic automata for periodic random environments. IEEE Transactions on Systems Man and Cybernetics 2 (2), 285–289.
- Maravall, D., J. New reinforcement algorithm in learning automata. Electronic Letters 16 (16), 623–624.
- D. Maravall, New Solutions in Adaptive Signal Detection, First Prize IEEE Region 8 PSPC, Eurocon 80, Stuttgart, 1980.
- S. Piskorski, N. Brulezi, F. D'Haeyer, AR.Drone Developer Guide, SDK 2.0, Parrot, 2012.
- Martin, J.A., de Lope, J., J. Learning autonomous helicopter flight with evolutionary reinforcement learning. In: Computer Aided Systems Theory. In: Moreno-Diaz, R., Pichler, F., Quesada, A. (Eds.), LNCS, vol. 5717. Springer Verlag, pp. 75–82.
- de Lope, J., San Martin, J.J., Martin, J.A., OJ. Helicopter flight dynamics using soft-computing models. In: Computer Aided Systems Theory. In: Moreno-Diaz, R., Pichler, F., Quesada, A. (Eds.), LNCS, vol. 4739. Springer Verlag, pp. 621–628.
- Barlow, J., 1J. The Cerebellum and Adaptive Control. Cambridge University Press.
- Imamizu, H., Miyauchi, S., Tamada, T., Sasaki, Y., Takino, R., Pütz, B., Yoshioka, T., Kawato, M., 2J. Human cerebellar activity reflecting an acquired internal model of a new tool. Nature 403, 192–195.
- Xu, G., Zhang, Y., Ji, S., Cheng, Y., Tian, Y., 3J. Research on computer vision-based for UAV autonomous landing on a ship. Pattern Recognition Letters 30, 600–605.
- Rizzi, A., Duina, D., Inelli, S., Cassinis, R., 4J. A novel visual landmark matching for a biologically inspired homing. Pattern Recognition Letters 22, 1371–1378.