Including Efficient Object Recognition Capabilities in Online Robots: From a Statistical to a Neural-Network Classifier

Pedro J. Sanz, *Member, IEEE*, Raúl Marín, and José S. Sánchez

***Abstract*—For those situations in which the user wants to interact with the system by using, for example, voice commands, it would be convenient to refer to the objects by their names (e.g., "cube") instead of other types of interactions (e.g., "grasp object 1"). Thus, automatic object recognition is the first step in order to acquire a higher level of interaction between the user and the robot. Nevertheless, applying object recognition techniques when the camera images are being transmitted through the web is not an easy task. In this situation, images cannot have a very high resolution, which affects enormously the recognition process due to the inclusion of more errors while digitalizing the real image. Some experiments with the Universitat Jaume I Online Robot evaluate the performance of different neural-network implementations, comparing it to that of some distance-based object recognition algorithms. Results will show which combination of object features, and algorithms (both statistical and neural networks) is more appropriate to our purpose in terms of both effectiveness and computing time.**

***Index Terms*—Incremental learning, neural networks, object recognition, online robots.**

I. Introduction

REMOTE-CONTROLLED robots were developed in the 1940s and were used by trained experts. A new class of remotely controlled robots is now accessible on the Internet: the online robots. These allow users from all over the world to visit museums, tend gardens, navigate undersea, float in blimps, or handle protein crystals.

The first generation of online robots came into existence in 1994. This generation of Internet robots [3], [6]–[9], [15], [16] is mainly based on robotic arms or simple mobile robots that are directly controlled by human operators. In other words, a human is in the control loop. These telerobots operate within a well-structured environment with little uncertainty and have no local intelligence.

In contrast, research on the second generation of Internet robots [19], [21], [22] has recently begun to focus on autonomous mobile robots that navigate in a dynamic and uncertain environment. The key features of this generation of Internet robotic projects are their autonomy and reactive behaviors which enable them to navigate and cope with uncertainty in the real world.

Online robots involve controlling robots or devices from a web browser remotely and differ from traditional teleoperation in several aspects. In addition to the problems associated with time delay, supervisory control, and stability, online robots must be designed to be operated by nonspecialists through intuitive user interfaces and to be accessible 24 hours a day. New methods are needed for coordinating simultaneous users coping with large variations in demand and time delay, and for detecting and recovering from unsupervised errors. In fact, crucial issues like the interface usability [17] still remain partially unsolved.

New capabilities arise frequently with the introduction of well-known techniques from other research domains, such as Pattern recognition and machine learning. For example, in most of existing online robots, there is not an automatic object recognition module because input to the robot is simply performed by filling in forms or selecting commands. In this context, some research has been recently addressed to make the interaction easier between the user and the robot. In particular, some statistical learning algorithms have been used in the UJI online robot [13], [14] to automatically recognize the objects appearing in a scene.

In this paper, we extend a previous work [13] by considering the application of neural networks as an alternative to other statistical learning techniques (i.e., the minimum distance, the k is the nearest neighbors and the k are the nearest centroid neighbors rules). In general, these statistical methods have shown high performance (e.g., in terms of recognition accuracy), although the main deficiency refers to the response time. Correspondingly, the aim of using a neural-network classifier, instead of a statistical approach, in the object recognition module of an online robot will be to improve the response time maintaining the efficiency.

From now on, the paper is organized as follows. Section II briefly describes the UJI online robot and, in particular, its object recognition module. Section III compares the statistical learning techniques with the neural networks. Section IV describes the object descriptors used. Section V reviews previous statistical methods. Section VI introduces the incremental learning algorithm. Sections VII and VIII provide the experimental results, with statistical and neural networks, respectively. Finally, conclusions and further work are outlined in Section IX.

Manuscript received October 30, 2003; revised May 4, 2004. This work was supported in part by the Spanish Ministry of Science and Technology (CICYT) under Grants TIC2003-08496 and DPI2001-3801, and in part by the Fundació Caixa Castelló under Grants P1-1B2002-07 and P1-1B200315. This paper was recommended by Guest Editor P. J. Sanz.

P. J. Sanz and R. Marín are with the Computer Science Department, Jaume I University, Castellón E-12071, Spain (e-mail: sanzp@uji.es; rmarin@uji.es).

J. S. Sánchez is with the Programming Languages Department, Jaume I University, Castellón E-12071, Spain (e-mail: sanchez@uji.es).

Digital Object Identifier 10.1109/TSMCC.2004.840055

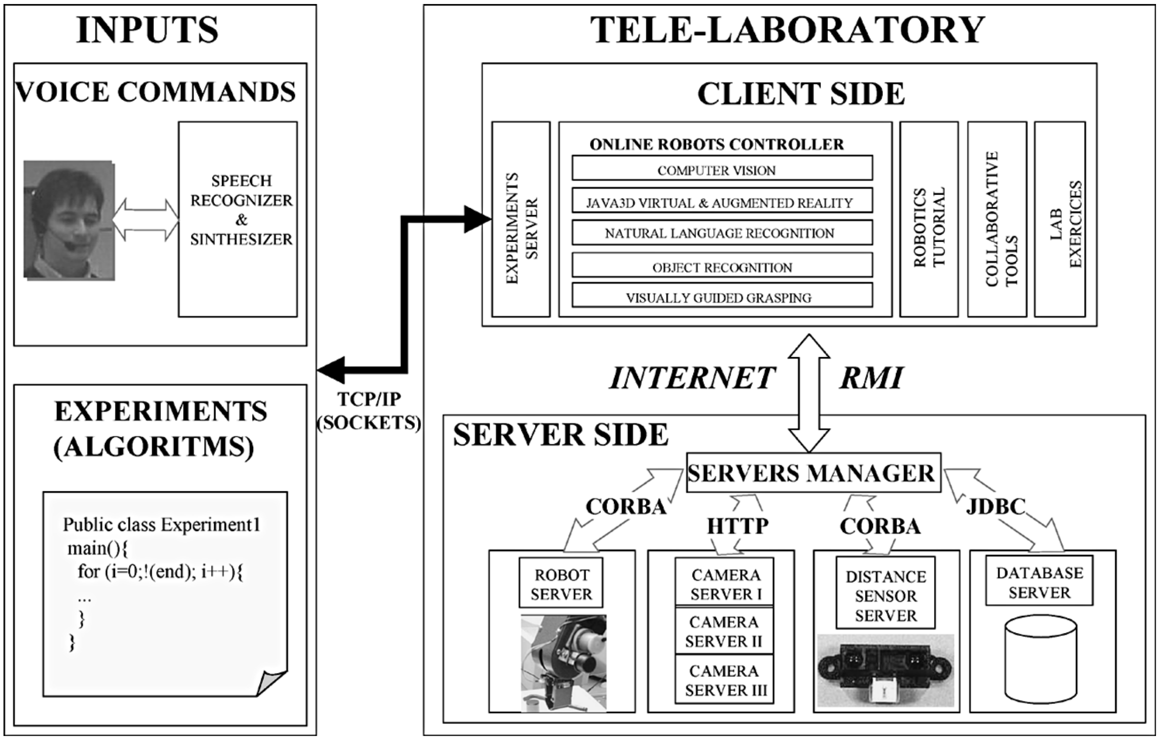


Fig. 1. System software architecture.

II. Short Description of the UJI Online Robot

The UJI online robot [13], [14] allows the manipulation of objects located on a board by means of mouse interactions and also by using a subset of the natural language. As the program is able to learn new object characteristics through the user interaction, the system becomes more robust as time goes by. Such a capability has not been reported in the frame of telerobotics yet.

The system is organized in a modular manner, involving different tasks: object recognition, natural language understanding, image processing, grasping and camera operations, etc. As can be seen in Fig. 1, the software architecture is organized into several modules connected through the CORBA and HTTP standards. The robot knowledge is organized into a database managed by the server side and accessed by the multiple Java clients running over the Internet. This means that the robot knowledge is common to the multiple users and even more important, it is robot independent.

In particular, the robot server is in charge of setting up both the robot and the camera, and also of controlling them. This subsystem runs on a PC and interacts with a SCARA robot via the parallel port. On the other hand, the user interface consists of a Java-based application that allows obtaining images of the robot workspace and sends commands to the robot server in order to control the robot and also to update the object database.

Unlike most of existing web-based robotic systems [3], [6]–[9], [15], [16], [19], [21], [22], the UJI online robot provides an object recognition module (Fig. 1), which allows the user to interact at a higher level.

This avoids the well-known cognitive fatigue, associated with the majority of online robots. In fact, as far as we know, apart from the UJI online robot, the system developed in the University of Columbia is the only one with automatic object recognition capabilities.

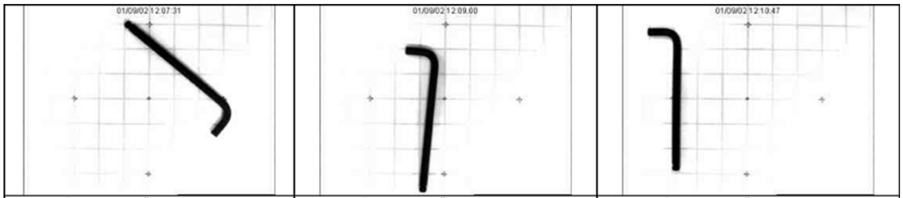
III. Statistical and Neural Object Recognition Approaches

Numerous taxonomies for classification methods in pattern recognition have been presented. One commonly stated division separates neural and more traditional statistical techniques [20] and, in this sense, a lot of benchmark and comparison studies [1], [10] have been published to illustrate the main differences between both classification perspectives.

From a practical point of view, the performance of an object recognition module in online robots must be measured in terms of two competing goals: recognition accuracy (or efficiency) and computing time. While the accuracy represents a check on the ability of the algorithms to correctly recognize objects, the computing time becomes a crucial requirement for real-time applications (this is the case, for instance, of online robots).

The statistical learning algorithms show, in general, a good performance in terms of efficiency, although they also present some important drawbacks related to the large response time. For example, the well-known k nearest neighbor decision rule [4] constitutes a widely used classifier because it does not rely on a parametric model, is extremely simple to implement, and asymptotically errs with a bounded rate that is at most twice the Bayesian error [2]. Nevertheless, the time required to classify a new object is high since this time is proportional to the number of features and the number of examples in a training set.

Taking into account the aforementioned disadvantage of most statistical learning methods (e.g., the k nearest neighbor), the neural approach appears to be a good alternative because of its



Sample "*allen01.jpg*" from the training set | Sample "*allen10.jpg*" from the training set | Sample "*allen19.jpg*" from the training set

Fig. 2. Samples of class "allen": their rotation, scale, and mirroring varies considerably.

TABLE I
HU DESCRIPTORS USING $f(x, y) = 1$ WHEN THE PIXEL BELONGS TO THE OBJECT'S SURFACE, 0 OTHERWISE

Surface	Hu1	Hu2	Hu3	Hu4	Hu5	Hu6	Hu7
Allen01	1,622449	2,794189	1,87E+07	1,90E+07	2,01E+14	1,48E+07	-3,15E+14
Allen10	1,714437	2,578057	6386389	6544710	-1,25E+13	6145301	2,50E+13
Allen19	1,661051	2,497977	1296658	1333882	1,75E+12	2085977	1,76E+12

TABLE II
HU DESCRIPTORS USING $f(x, y) = 1$ WHEN THE PIXEL BELONGS TO THE OBJECT'S BORDER, 0 OTHERWISE

Borders	Hu1	Hu2	Hu3	Hu4	Hu5	Hu6	Hu7
Allen01	9,907682	95,92047	1,14E+09	1,16E+09	1,33E+18	-1,51E+08	-1,33E+18
Allen10	8,627161	58,70144	54541,79	7825,882	1,65E+07	-55226,36	1,20E+08
Allen19	8,339341	59,98598	25256,68	1226,679	-4523629	-2527,641	-4113688

good performance in both efficiency and response time. In general, neural networks are superior in classification time to statistical techniques with similar or even better recognition accuracy.

Accordingly, in this paper, we study the employment of neural networks for object recognition in online robots. Despite the high efficiency of the statistical approaches used in our previous work [13], neural networks may provide similar recognition accuracy rates, but significant lower response time.

IV. OBJECT DESCRIPTORS

Basically, representing a scene object involves two choices.

1) Represent the object in terms of its external characteristics (its boundary).
2) Represent it in terms of its internal characteristics (the pixels comprising the region).

Choosing a representation scheme, however, is only part of the data useful to a computer. The next task is to describe the region based on the chosen representation. For example, a region may be represented by its boundary (e.g., perimeter), the orientation of the straight line joining the extreme points, and the number of concavities in the boundary.

Generally, an external representation is chosen when the primary focus is on shape characteristics. An internal representation is selected when the primary focus is on reflective properties, such as color and texture. In any case, the features selected as descriptors should be as insensitive as possible to variations such as changes in size, changes in location, and rotation [12].

In our case, the object recognition procedure is based on shape characteristics. Information such as color and texture have not been studied yet. The set of invariant descriptors that we have initially selected are listed below:

1) Hu descriptors (surface and borders approaches);
2) thinness ratio (Tr);
3) shape elongation (Lv);
4) spreading (Rv);
5) compactness (C).

A. Hu Descriptors

The Hu descriptors are a mathematical representation of a shape that has the particularity of being invariant to scale, rotation, and translation. Besides this, they are quite simple to obtain.

In order to calculate the invariant Hu descriptors associated with an object [11], it is necessary going through the central moments (invariant to the object location) and the normalized central moments (invariant to scale). Finally, we obtain the seven Hu moments that are invariant to location, scale, and rotation of the objects on the scene.

As introduced previously, this set of descriptors has the characteristic of being invariant to location, rotation, and scale changes. Moreover, in order to obtain a greater stability of the recognition procedure, 14 Hu descriptors can be used. The idea is to compute first the Hu descriptors for the pixels belonging to the object surface. It means that f(x,y) is considered 1 when the pixel belongs to the surface and 0 otherwise. The rest of descriptors are obtained considering the external contour of the object. For this situation, f(x,y) is 1 when the pixel belongs to the border and 0, otherwise.

It is easier to understand this situation by means of an example. Just consider the following set of samples for the same object class (i.e., the "Allen wrench" in Fig. 2).

The corresponding seven Hu descriptors for the surface are reported in Table I.

On the other hand, the calculated Hu descriptors for the border (i.e., external contour) are seen in Table II.

At this point, we could think about the following *question: Which of the 14 Hu descriptors is going to be more useful for* *recognizing an object?* Obviously, those values which are the most *stable* (variance close to 0.0) through the samples of a given class (e.g., "*Allen wrench*"). Moreover, being m_{ki} the average of descriptor i for class k, if the variance between m_{ki} for every class k in the training set is high, then the classification hits would be increased enormously.

By looking at Tables I and II, we can see how some of the Hu descriptors present more stability than the others. This is the case for Hu1 and Hu2, whose values for a set of samples of the same object present very little variation. Obviously, this should be taken into account when applying a recognition algorithm to the samples, as we will see in the next section.

B. Thinness Ratio Descriptor

A very useful and frequent descriptor is the measure of thinness, defined as [5]

$$T = 4\pi \left(\frac{S}{P^2} \right).$$

The S (i.e., area) and P (i.e., perimeter) values are real measures that must be approximated with a certain parameter once captured with a camera and digitally processed. In our case, the S parameter is estimated as the number of pixels contained within the object border.

A famous theorem is that T has a maximum value of 1, which is achieved if the figure in question is a circle. Analogously, from all possible triangles, the equilateral triangle has maximum T (of $T = \pi\sqrt{3}/9$), and from all quadrilaterals, the square has maximum T (of $T = \pi/4$).

Loosely speaking, then, the fatter a figure is, the greater the associated thinness ratio will be; conversely, line-like figures will have a thinness ratio close to zero. Moreover, the thinness ratio is dimensionless and, hence, depends only on the shape of the figure.

C. Elongatedness Descriptor

The elongatedness descriptor (L_v) is derived from the calculation of the best fit ellipse, whose maximum $(I_{\max})$ and minimum $(I_{\min})$ axes length are defined as [12]

$$I_{\min} = \frac{\mu_{20} + \mu_{02} - \sqrt{4\mu_{11}^2 + (\mu_{20} - \mu_{02})^2}}{2}$$
$$I_{\max} = \frac{\mu_{20} + \mu_{02} + \sqrt{4\mu_{11}^2 + (\mu_{20} - \mu_{02})^2}}{2}.$$

And then, the L_v descriptor is defined as

$$L_v = \frac{I_{\min}}{I_{\max}}.$$

D. Spreadness Descriptor

As in the previous case, the spreadness descriptor (R_v) is derived from the calculation of the best fit ellipse [12] too

$$R_v = \frac{I_{\min} + I_{\max}}{m_{00}^2}.$$

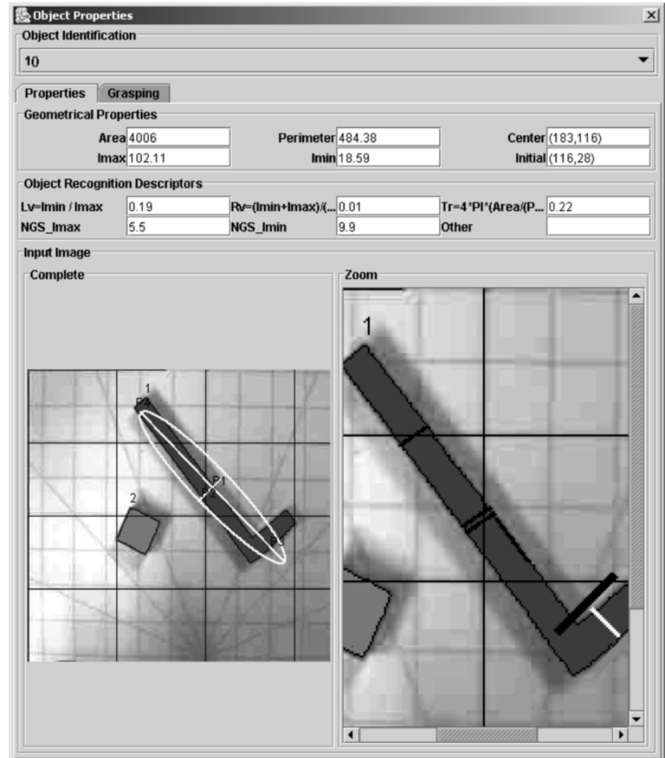


Fig. 3. Object properties window at the UJI online robot system.

E. Compactness Descriptor

Another descriptor based on the perimeter (P) and the area (S) refers to the compactness ratio (C), defined as [12]

$$C = \frac{P^2}{S}.$$

Compactness is a dimensionless quantity and, thus, insensitive to scale and translation changes. With the exception of errors introduced by the rotation of a digital region, compactness is also insensitive to orientation.

F. Object Properties Window

Within the UJI Online Robot 3-D virtual environment, once the object recognition procedure has been applied, the user is able to select an object in the scene and monitor its properties. By doing this, the "Object Properties Window" shown in Fig. 3 comes up.

As can be seen in Fig. 3, we are monitoring the object properties of the object "Allen wrench." The minus fit ellipse as well as the minimum and maximum inertia axis as represented in clearer gray over the object.

The "Object Recognition Descriptor" panel shows the most representative values for the object descriptors explained before. The "NGS_Imax" and "NGS_Imin" are part of an experiment that we are performing at this moment in order to use the "Normalized Global Symmetry" in order to test as an object descriptor.

V. Statistical Methods

Among recognition or classification techniques, those based on a form of distance measure probably constitute the most widely known methods. The popularity of these arises in part from their extreme conceptual and implementation simplicity and also in part from the fact that they model adequately a large number of practical situations. Within this context, the nearest-neighbor (NN) rule [4] is one of the simplest nonparametric classification algorithms devised, next only to the minimum-distance (MD) approach.

In the telerobotics system described in this paper, the object recognition module utilizes a distance-based scheme. With this end, a number of classifiers using several metrics have been tested in order to evaluate their performance when applied to a remote object recognition problem. In particular, the MD and k-NN decision rules, along with a recently proposed classification procedure, namely k nearest centroid neighbors (k-NCN) classifier, have been used.

A. MD Classifier

The MD classifier is arguably the most elemental nonparametric approach. Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of n previously labeled prototypes (namely, training set), and let $m_1, m_2, \ldots, m_c$ be the means for the given c problem classes. Then, a new sample x is classified by measuring the distance from x to each of the c means, and assigning x to the class for which the corresponding distance is minimum.

B. k-NN Classifier

NN methods have traditionally been used as an important pattern recognition tool in many different domains. In its classical manifestation, given an input sample x and a training set $X = \{x_1, x_2, \ldots, x_n\}$, the NN rule assigns any given sample to the class indicated by the label of the closest prototype in the training set. More generally, the k-NN rule maps any sample to the problem class most frequently represented among the k closest neighbors. Reader can refer to [4] for a complete survey of NN techniques.

An important property of NN methods refers to the fact that if the number of training prototypes is large enough, the error probability for the NN rule is asymptotically (that is, in the infinite sample case) at most twice that of the optimal Bayes classifier. Furthermore, the asymptotic performance of the k-NN rule is even better than that of the simple NN and a number of interesting bounds have been derived [2], [5].

C. k-NCN Classifier

Experience has shown that the theoretical asymptotic performance of the k-NN classification rules is not always possible. In practice, the number of samples available is not large enough and then the error rates can be too far from the expected optimal behavior. In accordance to this fact, many other models have been proposed in the last years as a way of improving the results of NN techniques on a range of practical problems.

On the lines of alternatives to NN classifiers, the k-NCN decision rule has been defined [9]. This scheme makes use of a neighborhood concept with two complementary constraints. First, the neighbors of a given point p should be as close to it as possible. Second, those neighbors should be also located as symmetrically around p as possible. Algorithmically, these NCNs can be obtained by the following [10].

1) The first NCN of a given point p corresponds to its NN, say q_1.
2) The ith neighbor, q_i, $i \geq 2$, is such that the centroid of this and all previously selected NCNs, $q_1, \ldots, q_{i-1}$ is the closest to p.

Note that this iterative procedure clearly does not minimize the distance from the input point to the centroid because it gives precedence to the individual distances instead. The proximity of the k NCNs to the given point is guaranteed due to the incremental nature of the way in which they are obtained from the first NN.

This kind of neighborhood can be further used to define the aforementioned k-NCN classifier. Thus, the k-NCN decision rule assigns any given sample to the problem with a majority of votes among its k-NCNs. A more detailed description of a number of k-NCN schemes can be found in [11].

D. Distance Measures

In this section, the different distance functions chosen for combining the classification procedures are introduced. In particular, the distance measures here studied correspond to Euclidean (D1), Mahalanobis (D2), normalized (D3), extended Euclidean (D4), and per-class extended Euclidean (D5). It is to be remarked that D3, D4, and D5 measures will apply some kind of weighted Euclidean distance as further defined.

The most significant difference among these metrics refers to the definition of the corresponding weights. Let n be the number of elements that define a Hu descriptor and let w_i denote the weight applied to an element, where i designates the identifier of a component in the Hu descriptor array. Then, a generic weighted Euclidean distance can be written as follows:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} w_i (x_i - y_i)^2}. \tag{1}$$

The weights used by D3 [see (2)] correspond to the inverse of the variance for each class. This means that the per-class variance has to be updated at any time and, therefore, its computational cost will be higher. On the other hand, the D4 function has been designed to speed up D3 and is based on a preprocessing step previous statistical analysis to define the constant weights used by this metric. The weights to be applied are known in advance and it is not necessary to manage any per-class variance in order to implement the recognition procedure. Besides this, in order to perform the previous statistical study, we must know in advance the set of classes that will be entered into the system

$$w_i = \left(\sum_{k=1}^{N} \left(\frac{1}{s_{ki}} \right) \right) \cdot \left(\sum_{k=1}^{N} \left(\frac{1}{m_{ki}} \right) \right) \tag{2}$$

where N denotes the number of classes, while S_{ki} and m_{ki} are the variance and the mean of the samples belonging to class k, respectively.

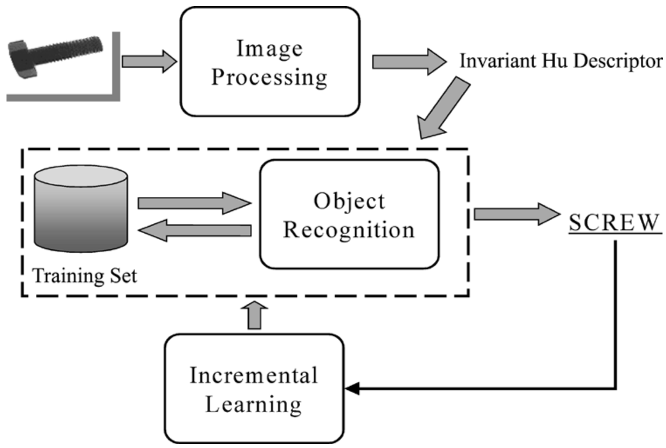


Fig. 4. Increasing knowledge of object recognition module.

Finally, the D5 measure [see (3)] has been designed as a way of including the D4 capabilities to a system where classes are not well known *a priori*. Thus, it defines different weights w_{ki} for each class k and then it takes into account scaling properties of each Hu descriptor as the D4 does

$$w_{ki} = \frac{1}{s_{ki}} \cdot \frac{1}{m_{ki}}. \tag{3}$$

VI. Incremental Learning Algorithm

The main goal of the present work is to make the object recognition module of our telerobotic system as automatic as possible, meaning that the system has to benefit from the experience obtained when working in the recognition of new examples. This incremental learning capability provides some nice advantages (Fig. 4): first, the recognition module will be more robust because errors in the training set can be corrected during operation and second, it enables the system to adapt to partially known or dynamic environments. As a consequence, it is expected that the performance will gradually improve over the lifetime of the telerobotic system.

On the other hand, taking into account that our telerobotic system operates with a web-based interface, an additional problem refers to the risk that anonymous users may add erroneous knowledge to the system, which might strongly degrade the performance of the recognition module. Accordingly, the incremental learning procedure implemented in our system has also the power of overcoming this difficulty.

The incremental learning algorithm here introduced covers a broad range of situations: objects belonging to new classes, mislabeled examples, atypical cases, and noisy data. Our method is based on simple distance-based classification rules and some other related techniques taken from the pattern recognition domain. In summary, the procedure consists of the following steps.

1) Let X be the initial training set, which is used by the object recognition module to classify new examples.
2) After recognizing a number of new examples, the system temporarily stops the object recognition module. As new examples are identified, they are stored in a set of candidate examples Y.
3) The system begins the learning process by checking the correctness of examples in Y prior to transferring them to X.
   3.1. The candidate examples in Y are classified by the k-NN rule (or any other classifier) with a reject option using the training set X: new object classes and mislabeled samples are detected.
   3.2. The Wilson's editing algorithm is employed as a second refinement to remove some of the examples moved from Y to X during the previous stage.
   3.3. If the number of examples in the current training set X is too large, the Hart's condensing algorithm is now applied so that the training set size is reduced.
4) Go to Step 2.

As can be seen, our approach to incremental learning focuses on improving the quality of training data by identifying and eliminating mislabeled examples prior to joining the current training set X and the candidate set Y. This is accomplished by applying two filters to the set of training examples: a classifier with a reject option and an editing procedure. The former aims at detecting new object classes and also mislabeled examples, whereas the latter deals with the problem of atypical instances.

Distinct reject options have been implemented in many classification models [13] as a way of reducing the misclassification rate of the system. In the specific case of our telerobotic system, the k-NN rule with the reject option has been defined as follows: if there is a majority of neighbors belonging to some object class, then the candidate example is assigned to that class and incorporated to X.

Otherwise, the example is added to the set X with its original class label. Note that our reject option has not the goal of eliminating examples, but finding new classes. Thus, the first filter of our learning algorithm is for detecting mislabeled examples and identifying new possible object classes.

The editing stage applied here corresponds to Wilson's editing [14], which consists of applying the k-NN classifier to estimate the class label of all examples in the training set and discard those whose class label does not agree with the class associated with the largest number of the k neighbors. In such a way, the learning algorithm is now detecting the atypical cases probably incorporated in the training set during the previous step.

Finally, the condensing procedure [15] can be applied as a way of reducing the current training set size. In brief, it consists of eliminating from the training set X those examples that are not necessary to correctly classify the rest of instances. This is a crucial stage in our incremental learning algorithm because the usually vast amount of training examples can prohibit its usage in all but the simplest of telerobotic domains.

VII. Experimentation With Statistical Methods

In this section, a comparative analysis of the classification algorithms described earlier using the metrics proposed in Section IV is presented. This study focuses on finding the object recognition scheme that provides the highest overall performance, that is, efficiency and recognition accuracy.

TABLE III
EXPERIMENT A: ERROR RATE AND COMPUTING TIME

Metric + Classifier	Error rate (%)	Computing time (msec)
D1 + MD	58.00	317
D3 + MD	46.67	322
D4 + MD	9.00	316
D5 + MD	43.66	420
D1 + *k*-NN	47.67	529
D3 + *k*-NN	41.67	565
D4 + *k*-NN	11.33	528
D5 + *k*-NN	41.67	864
D1 + *k*-NCN	43.33	1105
D4 + *k*-NCN	9.00	1125

TABLE IV
EXPERIMENT B: ERROR RATE AND COMPUTING TIME

Metric + Classifier	Error rate (%)	Computing time (msec)
D1 + MD	13.00	321
D3 + MD	30.10	324
D4 + MD	22.02	336
D5 + MD	45.08	341
D1 + *k*-NN	7.01	814
D3 + *k*-NN	22.50	860
D4 + *k*-NN	12.51	1085
D5 + *k*-NN	25.44	1232
D1 + *k*-NCN	11.50	1413
D4 + *k*-NCN	22.48	1428

The results here reported are averaged over five different random partitions of the original data set (i.e., half of the examples are for learning and half are for test purposes), to obtain the overall performances.

From each trial, computing time and error rate are calculated. The former gives a direct measure of the computational cost or efficiency associated to each alternative. On the other hand, the error rate provides a check on the ability of the algorithms to accurately recognize objects.

For the present experiments, various typical settings of the parameter k (neighborhood size), ranging from 1 through 11, have been tested and the one leading to the highest performance has finally been included in this section.

A. Well-Differentiated Objects Under Ideal Conditions

The first experiments have been carried out over a database generated from 120 images (with a resolution of 320 $\times$ 240 pixels) corresponding to six different objects (scissors, screwdriver, Allen wrench, tweezers, pliers, and screw), which determine the six problem classes. The 20 samples per class have been obtained by rotating, scaling, and translating the same object over the scene. In this case, the only object features considered are the 14 Hu descriptors (surface and border information) [15].

As can be seen in Table III, D4 metric combined with MD and k-NCN algorithms gives the lowest error rates, close to D4 used with k-NN rule. In fact, only these particular combinations seem to yield high performance in terms of recognition accuracy. Nevertheless, examining the other factor of interest, namely computing time, the results show that the k-NCN approach is much more computationally intense than MD and k-NN classifiers due to its $\mathrm{O}(kn)$ expected complexity [9] to search for k neighbors of a sample in a set of n points.

Note that D3 and D5 distance measures have not been applied to the k-NCN procedure because this approach requires a more exhaustive analysis in order to select the object class representing the actual centroid. Analogously, the performance (that is, error rate and computing time) corresponding to the D2 metric has not been included in this section because of the very high computing time.

Exploring both issues of run-time performance and recognition accuracy jointly, the results show that the best alternatives correspond to the use of the D4 measure with MD and k-NN models, that is, the closest combinations to the origin (0 s, 0% error). On the other hand, the results for the k-NCN classification scheme suffer from the large computational cost for calculating the successive centroids; they obtain a very low error rate (for example, employment of D4 with k-NCN provides an error rate of 9%), but also consumes a considerable amount of computing time (about 112 s). Obviously, this can constitute an important drawback for the practical application of this recognition technique in its present form.

B. Similar Objects Under Normal Conditions

In a web-based application with learning capabilities, the most common situation consists of having many kinds of objects, some of them presenting similarities and, therefore, classification can become a hard problem (in fact, some objects can define overlapping class regions).

On the other hand, lightning is controlled by means of a local lamp that illuminates the robot environment. However, depending on the time of day and even whether people are working or not at the laboratory, lightning varies accordingly. It means that the object segmentation procedure is merely affected by this situation, originating some little variations on the real object shape, and, consequently, on the object recognition result.

The present experiment tries to evaluate the robustness of the object recognition module under these real conditions (object similarity and lightning variations). Thus, for this second experiment, four different objects (Allen wrench, circle, cube, and Lego wheel) have been considered, and 20 samples per class have been obtained with varying lightning conditions. The features used in this experiment correspond to the thinness ratio, the shape elongation, and the spread factor.

Table IV shows that the D1 metric used with the k-NN and k-NCN classifiers achieves the lowest error rates, while computing times are similar to those obtained in the first experiment. It is worth mentioning that other experiments by using different combinations of features have been also tried but in most cases they have provided very poor recognition rates.

VIII. EXPERIMENTS WITH NEURAL NETWORKS

After a satisfactory experience with different statistical classification techniques [13], now a more elaborated contribution based on neural networks is presented. The ultimate aim of this is to evaluate the possibilities of using an alternative to those statistical schemes for object recognition in the online robots context where the response time is, in general, as important as the recognition rate.

Thus, the first step consists of calculating 100 examples for each object that will be used in the recognition phase. In our experiments, we have picked up four different objects or classes

(Allen wrench, cylinder, cube, and wheel), giving a total of 400 samples. After this, the set of 400 samples is divided into three different groups: training (200 samples), validation (100 samples), and test (100 samples).

Because of the existence of these four classes (Allen wrench, cylinder, cube, and wheel), the network will be provided with four neurons (s1, s2, s3, and s4) in its last layer. The s1 neuron will classify the sample as belonging to the Allen wrench object, the s2 as circle, the s3 as cube, and the s4 as wheel. The range of values that will be offered by each one of them is [0.1–0.9]. Thus, the outputs of the neural network will be (n1, n2, n3, n4), being n1 the result of the s1 neuron, n2 of the s2, and so on. The interpretation of any output from the neural network (i.e., n1, n2, n3, n4), can be understandable observing Fig. 5.

The procedure for training the neural network calculates the neuron weights according to the particular algorithm that is being used. The algorithms that we have tested in the experiments are the following.

1) Trainrp. resilient backpropagation [18];
2) Trainscg. scaled conjugate gradient backpropagation;
3) Traincgf. conjugate gradient backpropagation with Fletcher–Reeves updates;
4) Traincgb. Conjugate gradient backpropagation with Powell–Beale restarts;
5) Trainoss. One step secant backpropagation.

By using these five algorithms, the neural-network training will be accomplished when any of the following conditions is fulfilled.

1) We have reached the maximum number of iterations.
2) We have exceeded the time limit.
3) The efficiency has been minimized to the objective established.
4) The efficiency gradient is less than the minimum gradient.
5) The efficiency of the validation has been increased more than the maximum number of validation errors from the last time that was decreased.

On the other hand, related with the kind of descriptors used, note that we need to select a set of those mathematical descriptors that will enable the system to decide on which class of objects a real sample (as detected by the camera) belongs to. The idea is that the values of these descriptors are to be quite similar when the samples belong to the same class, and quite different when the samples belong to distinct classes. Thus, the selected descriptors are the invariant Hu descriptors (both surface and border alternatives) [11], thinness ratio (Tr), shape elongation (Lv), and spreading (Rv) [5].

A. Discussion

First, it is necessary to remark that all of the results have been obtained by using the *Neural Networks Toolbox of Matlab 6.1* [23] as a way to prototype in a very fast manner, with the end of isolating a software module in a further stage after conveniently analyzing these preliminary results.

A first experiment, with the aforementioned four classes, employs the whole set of object descriptors (i.e., surface and border Hu, thinness ratio, etc.) in order to train and test the corresponding performance by the neural-network classifier. These results are shown in Table V.

```
If ((n1 > 0.6) and (n2, n3, n4 < 0.4)) Then
 Allen wrench
Else
 If ((n2 > 0.6) and (n1, n3, n4 < 0.4)) Then
   Cylinder
 Else
  If ((n3 > 0.6) and (n1, n2, n4 < 0.4)) Then
    Cube
  Else
   If ((n4 > 0.6) and (n1, n2, n3 < 0.4)) Then
      Wheel
   Else
      Indetermination
```

Fig. 5. Algorithm to clarify any output from the neural network.

TABLE V
EFFICIENCY TAKEN INTO ACCOUNT THE WHOLE SET OF DESCRIPTORS

Topology	[10 4]	[50 4]	[10 10 4]	[100 100 4]	Average
Trainrp	92,20	94,20	93,00	95,70	93,78
Trainscg	94,40	94,20	92,90	92,20	93,43
Traincgf	84,30	87,00	84,00	89,20	86,13
Traincgb	92,20	88,40	92,00	92,90	91,38
Trainoss	91,60	89,40	93,80	93,60	92,10
Average	90,94	90,64	91,14	92,72	

We can see that some combinations of algorithm and topology obtain an efficiency rate of higher than 95% (i.e., Trainrp and 100;100;4). Besides this, we have realized the following situations:

1) When increasing the number of neurons in the intermediate layers, performance increases too.
2) For those applications where the response time is very important (e.g., online robots), we could select a partially complex configuration that yields good enough efficiency (i.e., 10;4, Trainrp). Moreover, for those applications where the time is not critical, we could select a more complex configuration (e.g., 100;100;4, Trainrp), which improves performance very much.

From a second experiment, Table VI reports the results when only the set of Hu surface descriptors is considered. In such a situation, one can observe that these descriptors are not sufficient to correctly discriminate the four problem classes. Thus, it is possible to conclude that this set of Hu surface descriptors is not good enough to identify a class in our scenario.

Now, we consider only the set of Hu border descriptors, and the results are shown in Table VII. Comparing with the averages in topologies and training algorithms in Table VII, it is obvious to conclude a worse performance by using the Hu border descriptors than by means of the Hu surface ones.

The preliminary conclusion would be the following: if neither Hu surface descriptors nor Hu border ones are good descriptors, the set of descriptors that must be good are the rest, that is, those that are related with Tr, Lv, and Rv descriptors. Thus, we can see these new results by using this set of three descriptors in Table VIII.

To complete the experiments, some results related to the response time are shown in Table IX, taking into account that all of the times, independently on the training algorithms and the

TABLE VI
EFFICIENCY WHEN ONLY HU SURFACE DESCRIPTORS ARE CONSIDERED

Topology	[10 4]	[50 4]	[10 10 4]	[100 100 4]	Average
Trainrp	75,2	75,7	74,1	76,5	75,38
Trainscg	73,7	73,4	72,9	78,2	74,55
Traincgf	72,1	66	66,5	74,7	69,83
Traincgb	70,1	64,9	58,1	76,1	67,30
Trainoss	73,1	74,7	74,6	76,1	74,63
Average	72,84	70,94	69,24	76,32	

TABLE VII
EFFICIENCY (%) WHEN ONLY HU BORDER DESCRIPTORS ARE CONSIDERED

Topology	[10 4]	[50 4]	[10 10 4]	[100 100 4]	Average
Trainrp	60,8	66,4	64,4	63,3	63,73
Trainscg	60,5	60,3	55,8	59,4	59,00
Traincgf	52,1	50,2	53,3	49,9	51,38
Traincgb	58,2	53,5	59	60,8	57,88
Trainoss	54,2	60,3	57	59,4	57,73
Average	57,16	58,14	57,90	58,56	

TABLE VIII
EFFICIENCY USING THE DESCRIPTORS SET: Tr, Lv, Rv

Topology	[10 4]	[50 4]	[10 10 4]	[100 100 4]	Average
Trainrp	90,7	92,4	90,8	93,5	91,85
Trainscg	86	92,4	91,4	92,1	90,48
Traincgf	80,2	86,1	85,2	92,5	86,00
Traincgb	86,3	91,9	91,9	93	90,78
Trainoss	90,4	90,7	91,4	95,2	91,93
Average	86,72	90,70	90,14	93,26	

TABLE IX
RESPONSE TIME FOR DESCRIPTORS: Tr, Lv, Rv

Topology	[10 4]	[50 4]	[10 10 4]	[100 100 4]
Trainrp	44	49	44	94
Trainscg	39	50	44	87
Traincgf	33	50	44	88
Traincgb	44	49	44	88
Trainoss	44	49	44	88

set of descriptors, are inside the interval (33;94) ms, depending basically on the topology used, because more complex ones require more computation time. These times have been obtained with an Intel Celeron to 400-MHz processor.

Related to the response time required for the training period, we can conclude that all of the times are inside the interval (20;70) s, including the more complex topology used. More specifically, the lower time corresponds to the Trainrp algorithm, while the higher is for the Trainoss one. It should be pointed out that always the vector of descriptors is taken as input.

IX. CONCLUSIONS AND FURTHER WORK

In this paper, we have studied the possibility of using a neural-network approach to object recognition in online robots, and discuss the main advantages over the application of statistical learning methods. Several experiments have been carried out with the UJI online robot in order to evaluate the performance of different neural-network implementations.

After analyzing the results shown in the corresponding tables, we can draw some interesting conclusions. The best algorithm performance is for Trainrp, developed by Martin Riedmiller. This algorithm achieves more than 95% of efficiency when a topology of (100,100,4) is used jointly with the complete set of descriptors, and more than 93% when the set of descriptors is limited to (Tr, Lv, Rv) for the same topology. On the other hand, another important issue is that when time is critical, an interesting possibility will be to employ this set of only three descriptors (Tr, Lv, Rv), jointly with the Trainrp algorithm, because the computational time to extract only these three descriptors is always lower than when the complete set is processed. Finally, another important result is that the border and surface Hu descriptors have not important discrimination capabilities for this case.

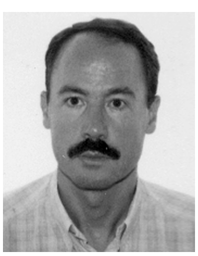
In our previous work with statistical learning methods [13], the best performance was achieved by a k nearest neighbor classifier (93% of efficiency, and response time of 814 ms). Note that under the same conditions (set descriptors Tr, Lv, Rv), the results from a neural network using Trainrp with the most complex topology (100;100;4) are even better (93.5% of efficiency, and response time of 94 ms). Thus, these results aim at following to investigate neural networks applied to online robots.

Related to possible application of the object recognition techniques to online robots in real scenarios, it is interesting to say that recently we are trying to extend our work to the humanitarian demining context. A first stage of consideration would be the possibility of including any kind of object reconstruction (i.e., the landmine), that is, partially occluded by the forest, ground, etc., assisting the user in his or her identification tasks. This could be a good example of augmented reality in real-life applications, namely, supported from object recognition capabilities.

REFERENCES

[1] B. Cheng and D. Titterington, “Neural networks. A review from a statistical perspective,” *Statist. Sci.*, vol. 9, no. 1, pp. 2–54, 1994.
[2] T. M. Cover and P. E. Hart, “Nearest neighbor pattern classification,” *IEEE Trans. Inform. Theory*, vol. IT-13, no. 1, pp. 21–27, 1967.
[3] M. J. Cox and J. E. F. Baruch, “Robotic telescopes: an interactive exhibit on the world-wide web,” in *Proc. 2nd Int. Conf. World-Wide Web*, Chicago, IL, 1994.
[4] B. V. Dasarathy, *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. Los Alamitos, CA: IEEE Comput. Soc. Press, 1990.
[5] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: Wiley, 2000.
[6] A. Ferworn, R. Roque, and I. Vecchia, “MAX: wireless teleoperation via the world wide web,” in *Proc. Int. Assoc. Science Technology Development Conf. Robotics & Applications*, Santa Barbara, CA, 1999, pp. 158–162.
[7] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and J. Wiegley, “Desktop teleoperation via the world wide web,” in *Proc. IEEE Int. Conf. Robotics Automation*, 1995, pp. 654–659.
[8] K. Goldberg, J. Santarromana, G. Bekey, S. Gentner, R. Morris, J. Wiegley, and E. Berger, “The telegarden,” in *Proc. Assoc. Computing Machinery Special Interest Group Computer Graphics Interactive Techniques*, 1995, pp. 135–1140.
[9] S. Goldberg *et al.*, “DIGIMUSE: an interactive telerobotic system for remote viewing of 3D art objects,” in *Proc. Intell. Robots System Workshop on Web Robots*, Victoria, BC, Canada, 1998, pp. 55–60.

[10] L. Holmström, P. Koistinen, J. Laaksonen, and E. Oja, "Neural and statistical classifiers—taxonomy and two case studies," *IEEE Trans. Neural Netw.*, vol. 8, no. Jan., pp. 5–17, 1997.
[11] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Inform. Theory*, vol. 8, pp. 179–187, 1962.
[12] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[13] R. Marín, J. S. Sánchez, and P. J. Sanz, "Object recognition and incremental learning algorithms for a web-based telerebotic system," in *Proc. IEEE Int. Conf. Robotics Automation*, Washington D.C., 2002, pp. 2719–2724.
[14] R. Marín, P. J. Sanz, and J. S. Sánchez, "A very high level interface to teleoperate a robot via web including augmented reality," in *Proc. IEEE Int. Conf. Robotics Automation*, Washington D.C., 2002, pp. 2725–2730.
[15] G. T. McKee and R. Barson, "NETROLAB: providing access to robotics technology using the internet," *Robotics Machine Perception. Special Issue: Networked Robotics*, vol. 5, no. 1, 1996.
[16] E. Paulo and J. Canny, "Delivering real reality to the world wide web via telerobotics," in *Proc. IEEE Int. Conf. Robotics Automation*, 1996, pp. 1250–1256.
[17] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, and T. Carey, *Human-Computer Interaction*. Reading, MA: Addison-Wesley, 1994.
[18] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the RPROP algorithm," in *Proc. IEEE Int. Conf. Neural Networks*, San Francisco, CA, 1993, pp. 586–591.
[19] P. Saucy and F. Mondana, "KhepOnTheWeb: open access to a mobile robot on the internet," *IEEE Robot. Autom. Mag.*, vol. 7, no. 1, pp. 41–47, Mar. 2000.
[20] R. J. Schalkoff, *Pattern Recognition: Statistical, Structural and Neural Approaches*. New York: Wiley, 1992.
[21] D. Schulz, W. Burgard, D. Fox, S. Thrun, and A. B. Cremers, "Web interface for mobile robots in public places," *IEEE Robot. Autom. Mag.*, vol. 7, no. 1, pp. 48–56, Mar. 2000.
[22] R. Simmons, "XAVIER: an autonomous mobile robots on the web," in *Proc. Intelligent Robots System Workshop On Web Robots*, Victoria, BC, Canada, 1998, pp. 43–48.
[23] [Online] Available: http://www.mathwoks.com



Pedro J. Sanz (M'98) received the B.Sc. degree in physics from the University of Valencia, Valencia, Spain, in 1985, the M.Sc. degree in engineering [compuer-aided design/computer-aided manufacturing (CAD)/(CAM)] from the Technical University of Valencia, Valencia, Spain, in 1989, and the Ph.D. degree in computer engineering from the Jaume I University, Castellón, Spain, in 1996.

Currently, he is Associate Professor of computer science and artificial intelligence, a Researcher with the Robotic Intelligence Lab, and Head of the Multimedia Research Group at Jaume I University, Castellón, Spain. He has been active in R&D within several projects on Advanced Robotics and Multimedia Interfaces since 1990. He is author or co-author of a broad range of research publications, collaborating meanwhile with different scientific societies such as IEEE R&A, IAPR, and ECCAI. His research interests include service robotics, particularly autonomous manipulation, telerobotics and human-robot interaction. He has been appointed Visiting Scientist at the Institute for Real-Time Computer Systems (Technische Universität München, Germany) periods in 2000 and 2001, and at Le Laboratoire des Sciences et Matériaux pour l'Electronique et d'Automatique, (LASMEA) with the University Blaise Pascal, France, in 2002.

Dr. Sanz is a member of the organizing committee of the 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA-AIE'98), Co-Chair of the Tutorial T1 ("Toward Intelligent Robotic Manipulation") within the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Co-Chair of an Interest Group ("Manipulation and grasping, lightweight manipulators") within EURON (European Robotics Research Network, 2001–2003). He was a 1996 recipient of the "Best Thesis in Electronics and Computer Science Domains," National Research Prize, from the "Artigas Spanish Foundation" (Technical University of Madrid, Madrid, Spain). He has served as reviewer of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS—PART A, as well as of several international conferences, including the International Conference on Robotics and Automation (ICRA2002, ICRA2004), 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), the International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems (IEA-AIE 2003), and the IASTED International Conference on Robotics and Applications (RA 2003), among others.

Raúl Marín received the B.Sc. degree in computer science engineering and the Ph.D. degree in engineering from the Jaume I University, Castellon, Castellon, Spain, in 1996 and 2002, respectively.

Currently, he is a Researcher in the Department of Computer Science and is a Lecturer on computer networking and telematics at Jaume I University. In 1996, he was a Software Developer with the Multimedia Department of BYG Systems Ltd., Nottingham, U.K. In 1997, he was a Researcher, Software Developer, and Software Architect with Lucent Technologies (Bell Labs Innovations) and was with the Switching and Access Division, Madrid, Spain. His research interests include object recognition, Internet telerobotics, multimedia, three-dimensional (3-D) virtual environments, and education tutorials. He is author or co-author of multiple research publications on these subjects. He has been appointed Visiting Scientist at the Laboratory "Le LAboratoire des Sciences et Matériaux pour l'Electronique et d'Automatique" (LASMEA), Univ. Blaise Pascal, France, in 2002.

Dr. Marín was a member of the 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA-AIE'98). He was Chairman of the EURON 2003 Symposium on Internet and Online Robots for Telemanipulation, and Guest Editor of the Special Issue on Telemanipulation within the *International Journal of Robotic Systems*. He has served as a reviewer of several international journals, such as the *Journal of Autonomous Robots*, *Special Issue on Internet and Online Robots*. Moreover, he has acted as Reviewer in several international conferences, such as the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) and the IASTED International Conference on Robotics and Applications (RA 2003).

José S. Sánchez was born in Tarragona, Spain. He received the B.Sc. degree in computer science from the Technical University of Valencia, Valencia, Spain, in 1990 and the Ph.D. degree in computer science engineering from Jaume I University, Castellon, Castellon, Spain, in 1998.

Currently, he is an Associate Professor in the Department of Programming Languages and Information Systems of Jaume I University, and is the Head of the Pattern Recognition Section in the Computer Vision Lab. He is author or co-author of many scientific publications and co-editor of two books. He has served as guest editor for the special issue on Advances in Pattern Recognition and Image Analysis of the *International Journal of Pattern Recognition and Artificial Intelligence*, and for the special issue on Pattern Recognition and Image Analysis in Cybernetic Applications in *Cybernetics and Systems.* His research interests include the pattern recognition domain, including classification, feature and prototype selection, ensembles of classifiers, fuzzy systems, and decision tree induction, along with pattern recognition applications to robotics.

Dr. Sánchez was a member of the organizing committee of the 8th Spanish Conference on Computational Geometry and Co-Chair of the 9th Spanish Symposium on Pattern Recognition and Image Analysis. He has served as reviewer of the international journals *Pattern Recognition Letters*, *Information Fusion*, and *Pattern Analysis and Applications*, as well as of several international conferences, including the World Multiconference on Systemics, Cybernetics and Informatics, and the International Conference in Central Europe on Computer Graphics and Visualization. He was the winner of the 1997 Best Predoctoral Work Award of the Spanish Association for Artificial Intelligence. He is a member of IEEE Signal Processing Society, IEEE Computational Intellignece Society, IEEE Information Theory Society, the International Association for Pattern Recognition (IAPR), Spanish Association of Pattern Recognition and Image Analysis (AERFAI), the European Coordinating Committee for Artificial Intelligence (ECCAI), and the Spanish Association for Artificial Intelligence (AEPIA).