Proceedings of the 2002 IEEE
International Conference on Robotics & Automation
Washington, DC • May 2002

Object Recognition and Incremental Learning Algorithms for a Web-based Telerobotic System

R. Marín, J. S. Sánchez, P. J. Sanz

Universitat Jaume I, Av. Vicent Sos Baynat s/n, E-12006 Castellón, Spain
E-mail: {rmarin,sanchez,sanzp}@uji.es

Abstract– This paper describes a web-based telerobotic system with two basic capabilities: object recognition and incremental or continual learning. Within this context, the present paper investigates the feasibility of applying several distance-based classifiers to remote object recognition tasks. On the other hand, incremental learning enables the system to maintain a representative set of past training examples that are used together with new data to appropriately modify the currently held knowledge. This can constitute a very significant advantage because it allows a system to continually increase its knowledge and accordingly, to enhance its performance in recognition as it learns. And also, because this capability allows to detect an incorrect or anomalous employment of this particular remote system. The performance of the system is evaluated by means of an empirical analysis addressed to assess two competing goals: computing time and recognition accuracy.

Keywords– Distance function, Incremental learning, Object recognition, Telerobotics, Web-based technology.

I. INTRODUCTION

Telerobotics could be understood as the ability to control a robot remotely. As explained in [1], telerobotics finds application to hazardous environments, control of underwater remotely operated vehicles, space systems, nuclear industry applications, training, entertainment, etc. Besides this, World Wide Web constitutes a widely accessible and low-cost interface that makes available some devices like robots to a broad range of users.

One of the first telerobotic systems with a web-based interface was presented in the Mercury project [2], developed by researchers from the University of Southern California. In this system, the manipulator was a robotic arm equipped with a camera and a compressed air jet, and the interface consisted of a web page accessible using any standard browser. The robot had to explore and excavate a sandbox full of artifacts. The interface allowed the user to move the robot to a point in the workspace and blow a burst of compressed air into the sand directly below the camera; all robot controls were available via the mouse interaction.

The University of Western Australia developed another telerobotic system [1], in which the user could control an industrial robot to manipulate objects distributed on a table. The user interface allowed a human operator to specify the coordinates of the desired position of the arm, the opening of its gripper and other multiple parameters by filling forms and clicking images through an HTML interface.

One of the fundamental points in the design of a complete web-based telerobotic system refers to the definition of the human-robot interaction [3]. In most systems, user interaction is still computer-oriented, because of input to the robot is accomplished by filling in forms or selecting commands from a panel. Nevertheless, up to now very little attention has been paid to more unrestricted ways of communication such as natural language or gestures. On the other hand, most telerobotic systems do not care about object recognition since this task is directly performed by a human operator.

The present work introduces a web-based telerobotic system with three relevant characteristics. First, the human-robot interaction is based not only on some mouse operations but also on a simplification of the natural language. On the other hand, the system consists of a complex object recognition module. Along with this, the telerobotic system has also the capability of an automatic incremental learning [4] when working in the recognition of new patterns, that is, the object recognition module will utilize the knowledge acquired from practice. This allows to enhance the recognition performance of the system.

Section II provides an overview of our web-based telerobotic system. Section III describes some classification algorithms applied to object recognition tasks. Section IV presents a set of distance functions employed with the classification approaches. Section V introduces an incremental learning procedure that allows the system to continually increase the knowledge utilized by the object recognition module. Experiments in Section VI evaluates the performance of the classification models combined with different metrics. Finally, conclusions and further work are outlined in Section VII.

II. Overview of the Telerobotic System

The telerobotic system [5] described in this paper 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. As introduced above, such a capability has not been reported in the frame of telerobotics, yet.

In brief, the overall telerobotic system is a flexible modular system 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.

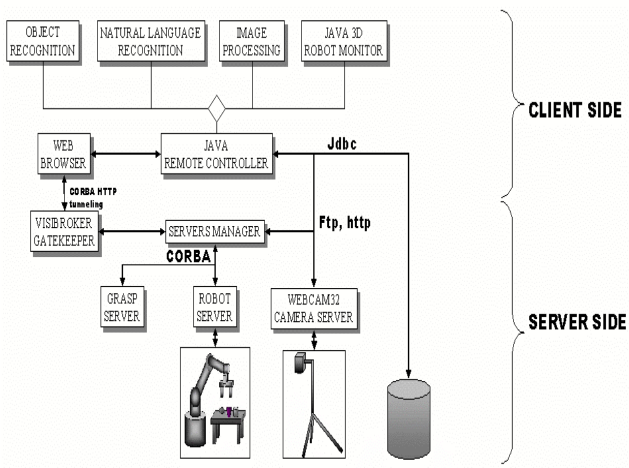

Fig. 1. System software architecture.

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 to obtain 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.

III. Classification Models

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 implementational 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 [6] is one of the simplest non-parametric classification algorithms devised, next only to the Minimum-Distance (MD) approach.

In the telerobotic 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. The MD Classifier

The MD classifier is arguably the most elemental non-parametric approach. Let $X = \{x_1, x_2, ..., x_n\}$ be a set of n previously labeled prototypes (namely, training set), and let $m_1, m_2, ..., m_c$ 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. The 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 $X = \{x_1, x_2, ..., x_n\}$, 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 [6] 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 [7].

C. The 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 farq 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 [8]. 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 as follows [9]:

1. The first NCN of a given point *p* corresponds to its NN, say q_1.
2. The *i*-th neighbor, q_i, $i \geq 2$, is such that the centroid of this and all previously selected NCNs, $q_1,..., q_{i-1}$, 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. 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 [8].

IV. DISTANCE MEASURES

In this section, the different distance functions chosen for combining with 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:

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

The weights used by D3 (see Eq. 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 pre-processing 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.

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

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.

Finally, the D5 measure (see Eq. 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.

$$w_{ki} = \frac{1}{s_{ki}} \cdot \frac{1}{m_{ki}} \quad (3)$$

V. AN 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 [10]. This incremental learning capability provides some nice advantages (see Fig. 2): 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.

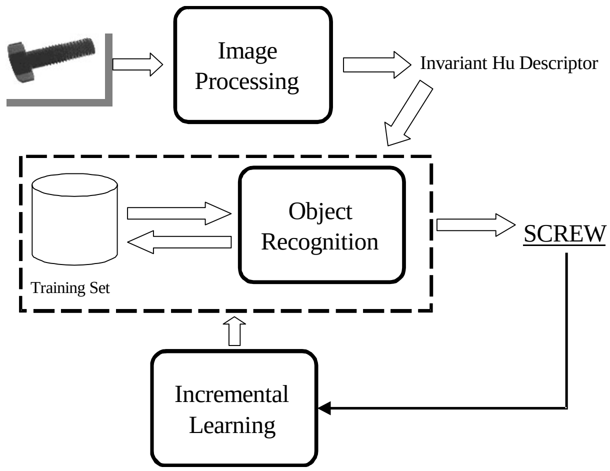

Fig. 2. Increasing knowledge of object recognition module.

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 [11] 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 [12], 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 to the training set during the previous step.

Finally, the condensing procedure [13] 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.

VI. EXPERIMENTATION

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.

The results here reported are averaged over five different random partitions of the original data set (i.e., half of examples for learning and half 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 320x240 pixels) corresponding to six different objects (scissors, screwdriver, Allen key, 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 fourteen Hu descriptors (surface and border information) [14].

As can be seen in Table I, 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 O(*kn*) expected complexity [9] to search for *k* neighbors of a sample in a set of *n* points.

TABLE I
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

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 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 sec, 0% error). On the other hand, as is to be expected from the algorithm described in Section III.C, 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 sec). 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 is 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 key, 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 II
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

Table II shows that the D1 metric used with the *k*-NN and *k*-NCN classifiers achieve 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.

VII. Conclusions and Further Work

This paper describes the application of pattern recognition and machine learning techniques to a web-based telerobotic system. Accordingly, the focus of the present paper is two-fold: improving the performance of the object recognition module by means of different classification techniques, and proposing an incremental learning algorithm that continually increases the knowledge of the telerobotic system.

With respect to the recognition task, a number of conventional and novel distance-based classification schemes, along with several metrics have been tested in the experimental section, searching for the one with lowest computing time and highest accuracy. From the set of experiments carried out in Section VI, some preliminary conclusions can be drawn. The *k*-NN and *k*-NCN classification algorithms generally achieve the highest recognition accuracy but they are also computationally more expensive than the MD classifier.

On the other hand, the combination of D4 (that is, the extended Euclidean distance) with the MD classifier generally produces the best results in terms of balancing computing time for implementation purposes with classification accuracy. However, it is to be mentioned that this combination would be suitable only for those systems in which the objects to manage by the robot are known in advance, which does not constitute a very common practical situation.

The incremental learning algorithm proposed here makes use of multiple filters in order to deal with different situations. First, the *k*-NN classifier with a reject option is employed as a way of identifying new object classes and also detecting mislabeled examples. Second, the editing scheme eliminates atypical samples. Finally, the usage of a condensing procedure allows to reduce the training set size in order to decrease complexity relative to handling a large number of examples in the training set.

Future plans include investigation of other families of instance-based classification models and also other metrics to achieve even better performance in terms of a well balanced trade-off between run-time and object recognition accuracy. A second direction in our further work focuses on including new learning capabilities to the telerobotic system. Within this context, the employment of unsupervised techniques could help to reliably detect new object classes among the candidate examples.

Acknowledgments

This work was supported in part by the Spanish CICYT (Department of Science and Technology) under Grant No. TIC2000-1703-C03-03 and by the Generalitat Valenciana under Grant No. GV01-244.

References

[1] K. Taylor and B. Dalton, “Internet robots: a new robotics niche”, *IEEE Robotics & Automation Magazine*, vol. 7(1), 2000.

[2] K. Goldberg, S. Gentner, C. Sutter, and J. Wiegley, “The Mercury project: a feasibility study for Internet robots: design and architecture of a public teleoperated system accessible by anyone on the Internet at any time”, *IEEE Robotics & Automation Magazine*, vol. 7(1), 2000.

[3] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, and T. Carey, *Human-Computer Interaction*, Reading, UK: Addison-Wesley, 1994.

[4] B. Natarajan, *Machine Learning: A Theoretical Approach*, Morgan Kaufmann, 1991.

[5] R. Marín and P. J. Sanz, “The UJI telerobotic training system”, in *Proc. 1st Workshop on Robotics Education and Training*, Weingarten, Germany, 2001, pp. 25-30.

[6] B. V. Dasarathy, *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*, Los Alamitos, CA: IEEE Computer Society Press, 1990.

[7] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*; New York: Springer, 1996.

[8] J. S. Sánchez, F. Pla, and F. J. Ferri, “Surrounding neighbourhood techniques for nearest-neighbour based classification and prototype selection”, in *Recent Research Developments in Pattern Recognition*: Transworld Research Network, 2000, pp. 63-76.

[9] B. B. Chaudhuri, “A new definition of neighborhood of a point in multi-dimensional space”, *Pattern Recognition Letters*, vol. 17, pp. 11-17, 1996.

[10] S. Thrun and T. Mitchell, “Learning one more thing”, in *Proc. 14th International Joint Conference on Artificial Intelligence*, San Mateo, CA: Morgan Kaufmann, 1995.

[11] P.A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice Hall: Englewood Cliffs, NJ, 1982.

[12] D. L. Wilson, “Asymptotic properties of nearest neighbor rules using edited data sets”, *IEEE Trans. on Systems, Man and Cybernetics*, vol. 2, pp. 408-421, 1972.

[13] P. E. Hart, “The condensed nearest neighbor rule”, *IEEE Trans. on Information Theory*, vol. 14, pp. 515-516, 1968.

[14] M. K. Hu, “Visual pattern recognition by moment invariants”, *IRE Trans. on Information Theory*, vol. 8, pp. 179-187, 1962.