VISUAL PATH FOLLOWING AND OBSTACLE AVOIDANCE BY ARTIFICIAL NEURAL NETWORKS

P. Burrascano[*], *S. Fiori*[*], *F.M. Frattale-Mascioli*[**], *G. Martinelli*[**], *M. Panella*[**], *A. Rizzi*[**]

[*]Dept. of Industrial Engineering, University of Perugia, Via Duranti 1/A-4, Perugia, Italy
[**]INFOCOM Department, University of Rome "La Sapienza", Via Eudossiana, 18, Rome, Italy
E-mails: {burrasc,sfr}@unipg.it; {mascioli,martin,panella,rizzi}@infocom.uniroma1.it

Abstract: The aim of this work is to propose an artificial neural network based solution to a robot guidance problem, namely to visual path following and obstacle avoidance. Robot guidance task is performed through neural algorithms that allow a controller to determine robot's location within an environment like a building and to re-plan the trajectory of the robot both in the case that small deviations from the prescribed one are observed and in case of large obstacles to be avoided. The visual information only is exploited, and the neural networks are used both to learn the robot control laws and to extract from the visual data the necessary cues for obstacle avoidance. In this paper we also propose a path planning procedure in presence of unknown obstacles, which are detected by distance sensors. We suppose that the robot knows its position as well as the target one. The path planning procedure employs two different algorithm: the surrounding algorithm (SA) and the navigation algorithm (NA). The first one is activated when the robot is close to an obstacle, while the second one is useful to reduce the length of the path.

Keywords: Multiplayer-perceptrons, relevance determination, optimal network pruning by generalization maximization, motion planning, clustering.

1 INTRODUCTION

Over the recent years, many dedicated neural-network based algorithms have been developed and applied to robot motion control by digital image processing. A partial list of relevant recent references is given by [Dickmanns and Mysliwetz, 1992; Karnin, 1990; Kluge, 1993; Kröse et al., 2001; Payton, 1992]. Such neural-network solutions have been envisaged basically in order to extract relevant information from complex image ensembles or sequences. Intelligent neural solutions are preferred over classical image processing techniques thanks to the high flexibility of the neural structures and because their synthesis does not require a complete knowledge of the problem model; conversely, the networks may learn the correct structure from the available data (in supervised learning) and may self-extract relevant information from data without any external cue (unsupervised learning).

Image processing by neural networks, however, presents a specific difficulty related to the high data dimensionality. It is evident that giving as input to a network directly the image to be processed would mean to associate an input neuron to each image pixel, leading to very complex networks also in the case of low dimensional images. Moreover, it is intuitively known that within a given image ensemble, the different support portions carry a different amount of information about the network task: in other terms, they possess different levels of relevance with respect to the mapping task at hand.

These considerations lead us to the conclusion that performing a pre-processing like image down-sampling is required in order to lighten the network topology or, equivalently, to prune network's inputs; furthermore, *this operation may be carefully performed by measuring the relevance of different portions of the image and retaining as useful inputs the ones coming from high relevance areas*.

Data-dimensionality and network-size reduction may be carried out in a problem-independent way through unsupervised neural networks (for a recent review see e.g. [Costa and Fiori, 2001; Kröse et al., 2001]) and in a problem-dependent fashion.

Among others problem-dependent techniques, network pruning by feature scaling/selection based on relevance determination relies on a strong theory recently developed [Battiti, 1994; Bollacker and Ghosh, 1996; Bishop, 1994; Burrascano, 1993; Granvalet and Canu, 1995; MacKay, 1992; Thodberg, 1991].

In the present work we propose to perform visual image down-sampling by training a neural network to extract some features from the original image ensembles, computed on a number of image support regions. The location and extension of these regions are to be defined by means of a

neural algorithm that is able to distinguish those parts of the image endowed with the highest relevance to the end of robot motion control. In particular, the proposed approach relies on relevance determination by generalization maximization [Burrascano, 1993].

Also, the development of efficient algorithms for path planning is fundamental for a great number of practical applications, such as the remote exploration of submarine environments and the design of automatic systems for transportation of goods. The presence of unknown obstacles represents an additional difficulty, since the path cannot be planned in advance; the robot movements must be decided step by step, by a suited set of rules, on the basis of the data coming from the sensors available on the robot [Lumesky and Stepanov, 1987; Yap, 1987]. The efficiency of the planning algorithm can be improved by employing neural networks, by which it is possible to model the relation between the vector of sensors' values and the movement direction. The model can be trained in an unsupervised manner by a clustering technique [Everitt, 1974], starting from an appropriate set of examples (training set).

In the present paper, the movement of the robot in an unknown environment in presence of obstacles is pursued by means of two algorithms[1]. The first one is denoted as "surrounding algorithm"(SA). It is activated when the robot approaches an obstacle; the activation distance is optimized with respect to the type of robot and the shapes of the obstacles to be encountered. The second algorithm is denoted as "navigation algorithm" (NA). It operates when the first one is not active.

The robot is equipped with devices which detect its current location and that of the target. We assume the environment to be enclosed in a known rectangle and containing objects of unknown shape and location. Two sides of the previous rectangle are chosen as the reference axes with respect to which the locations of the robot and of the target are defined.

2 FEATURE SELECTION ALGORITHM

The aim of the present section is to briefly explain the feature selection theory which leads to a learning algorithm for MLP-like neural networks. This learning algorithm allows the network to learn a path following task through visual image processing by the minimum-size network, that is the least complex network which still ensures the required network performance level.

As mentioned, our aim is to determine a network structure, starting from an assigned MLP network, as a trade-off between network structural complexity and input-output performance. A classical way to reduce the complexity of a neural structure is to progressively prune its nodes on the basis of some node-relevance determination criterion.

To this aim it is worth recalling that the classical performance measures for MLP-like learning are the *representation error* and the *generalization error*; the former criterion gives an information about the goodness of network learning on the training set, i.e. it gives an indirect measure of the adequateness of the neural structure to approximate the non-linear function which best explains the available data; conversely, the generalization error is a figure of specialization of the network, as it measures the network performances on a test set, independent form the training set. If a network has too many parameters (i.e. too many degrees of freedom), it usually learns much precisely the training input-output pattern association, but it also specializes excessively on these example and does not really form a model of the data, so that, when operated on different input-output patterns it fails the association. In other words, in this case the network does not generalize.

It follows that a network pruning procedure may be developed following the maximum-generalization principle.

Let us consider a feed-forward neural network trained on a number of input-output patterns. Its generalization ability with respect to a new input-output pattern may be defined as the probability that the network outputs the new response vector when presented with its corresponding input pattern; such probability computes as the probability of the input-output pattern-pair conditioned to a network state, averaged by the statistical distribution of all feasible network states.

In order to make this measure be of practical use, we may define an estimator of the generalization probability. The complete derivation of the generalization ability, on the basis of this probability function, is quite complex and is not reported here because of space limitations; it may be found in the previous

[1] Software C++ versions of the two algorithms are available at INFOCOM Department of the University "La Sapienza" of Rome.

contributions [Burrascano, 1993; Fiori and Burrascano, 2000; Fiori et al., 2001]. It can be briefly summarized as follows:

1. If the test set contains many statistically independent patterns, the generalization ability can be measured by the product of the single input-output pair correct association probabilities. A log-warping of this quantities gives an additive estimator, much easier to handle.
2. By using a fundamental inequality, arising from the theory of statistical learning [Tishby et al., 1990], we rewrite the generalization estimator as a function of average generalization error made by a network trained over the whole training set and on a number of test set;
3. Motivated from the possible problem of having a small-size data-set to train and validate the network, we cancel the distinction between training- and test-set: the effective test-set may be formed by discarding just one pattern at a time from the training set and using it to measure the performances of the network trained over the reduced-size training-set;
4. The ensemble-average quantities involved in the equation provided by the theory of statistical learning are not easily accessible; therefore the actual averages are replaced by sample mean obtained by averaging over the training-set samples and by repeating each operations many times (e.g. by learning over many epochs).

The result of such theoretical work is an additive generalization measure, that can be easily evaluated on the basis of the available data.

Of course, the generalization ability measure is a function of network parameters and, in particular, of network's weights. This suggests that it would be possible to extract some local information from the generalization global value by introducing a kind of *sensitivity* of generalization function which respect to any network parameter. The sensitivity is a meta-parameter assigned to each node and is termed *relevance* of the corresponding node.

After network learning, we can assign each node its corresponding relevance level and operate network pruning by simply ranking the collected relevance values and cutting the nodes having the least relevance figure.

An interesting aspect of the theory is that the sensitivity computation may be inherently performed during the network adaptation by back-propagation, because the quantities required for evaluating the relevance coincide to some quantities already made available by the error back-propagation procedure [Burrascano, 1993; Fiori and Burrascano, 2000]. This leads to an integrated learning/relevance-determination procedure termed G*eneralization-maximization Feature Selection* (GFS).

As the computation of the estimator requires directly available quantities that are elaborated by the back-propagation procedure during any learning iterations, the GFS algorithm is no more complex than back-propagation.

3. IMAGE DOWN-SAMPLING NET

The aim of neural network in our image processing problem is to model a function of the image, i.e. a non-linear mapping that represents the motion control law. To this aim we employ a multi-layer perceptron (MLP) network with one hidden layer of neurons, endowed with sigmoidal activation functions. The image coming from the camera should then input the network which give an approximation of the desired control navigation signal.

As mentioned, the image signal cannot directly enter the network, but it needs to be adaptively down-sampled, in order to dramatically reduce the size of data.

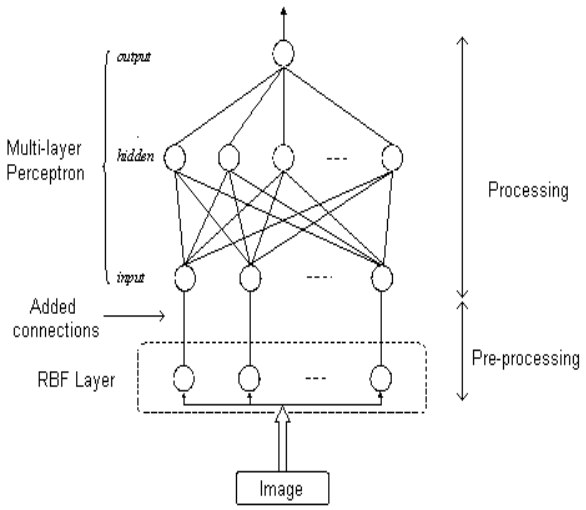

Figure 1. Schematic of the whole down-sampling neural network.

3.1 Details of image down-sampling

The image sampling operation may be formally represented by means of a non-linear operator that significantly reduces the size of the input data to low dimension data; the values resulting from

down-sampling operator based pre-processing may then input the neural network. Depending on the choice of the sampling operator, the reduced size network should be able to approximate again the true input-output mapping, ultimately making an acceptable approximation error.

The down-sampling strategy relies on the following operations:

1. The image support is partitioned into a pre-defined number of portions, approximately non-overlapping and covering the whole support. The portions have variable areas and locations, so that they can be moved and expanded or shrunken. Each portion is associated a value, depending upon the gray-value of the pixels within the support's portion and upon a weighting function, which measures the distance of each pixel from the center of the portion.
2. Starting from the initial sampling that uses a number of locations uniformly distributed over the whole image, by means of the GFS algorithm the network pruning is performed on the basis of inputs relevance determination; thus the number of inputs gets progressively reduced while the mean square representation error is kept below a pre-defined threshold. In this way, a number of sampling locations is retained; nevertheless, in general they are not yet optimally located;
3. The remaining sampling portions are then moved around the high relevance areas, by means of a proper strategy that takes into account the local properties of the image, and at the same time the non-linear sampling operator is varied by changing its parameters, in order to improve the mean square representation error.

It is necessary to specify the law with which a value is associated to each portion, namely, the structure of sampling operator.

3.2. Details of sampling operator

The first step in the theory is to build up a non-linear sampling operator. It is based on the concept of sampling kernel, and the whole structure will be implemented by the help of a radial-basis kernel neural network (RBF).

Each kernel is associated a centroid, i.e. the coordinates of its center on the image support, and a stretching parameter that controls its spatial extension. The kernel is a radial bell-shaped function, whose value only depends from the distance between the point it needs to be compute in and the kernel's centroid. By integrating the pixels' gray-values weighted by the kernel we obtain the down-sampled image value corresponding to that kernel; this operation is repeated for all down-sampling locations in order to obtain a virtual down-sampled image. A schematic of the whole down-sampling neural network is depicted in Fig. 1.

This is assumed as a valid parametric function of the image that gives different results depending on the positions of the centroids and the extensions of the kernels. The obtained RBF-derived down-sampled image can feed the MLP network, whose performance of course depends on the quality of down-sampling. Because of this functional dependence, both centroid and stretching parameters may be varied by standard gradient descent on the whole network generalization error.

4. EXPERIMENTAL RESULTS ON VISUAL PATH FOLLOWING

In order to assess the theoretical developments proposed in the previous sections, we present some experimental results obtained with real-world image sequences. Due to space limitations, we only briefly describe the experimental setup and obtained results; more detailed experiments may be found in [Fiori et al., 2001].

Figure 2. Example of the image set used in the experiments: Hallway where the pictures have been taken.

4.1 Experimental set-up

The used images were shot by means of a digital photo-camera in a hallway (see e.g. Fig. 2); the resolution was 140×105 gray-level pixels, and a total amount of 100 pictures was obtained. Of these, 50 pictures formed the training set or learning epoch. The robot motion control law is linear with dynamics [0.1, 0.9] and represents the

desired trajectory correction with respect to the center of the hallway (the value 0.5 means that no correction is necessary).

In the experiments, the image support was partitioned into 49 down-sampling areas, thus the employed neural network had 49 inputs, initially. Also, the kernel were chosen as Gaussian-bells, and the stretching parameter was therefore the 'variance' parameter.

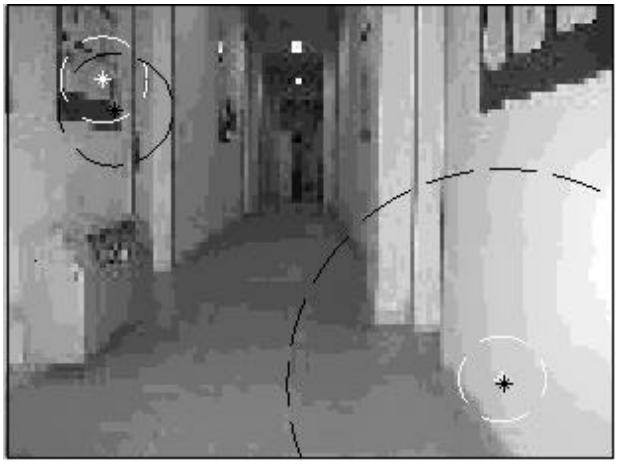

Figure 3. Final appearance of the two survived kernels on a sample image (white circle: sampling areas after kernel adaptation; black circles: sampling areas after kernel adaptation).

4.2 Experimental results

The relevance profile of the network's inputs evaluated on the basis of the 50 hallway images is not uniform, thus the different parts of the images carry on different amount of information.

During MLP's learning with input relevance determination, the progressive elimination of 45 sampling locations was achieved.

Then, according to kernel adaptation algorithm, the centroids of the four survived kernels were moved and stretching parameters were modified in order to improve network performances.

The whole learning phase was executed again: this led to the elimination of two more kernels without growing the representation error. The achieved error is about 10^{-4}, and the configuration of the kernels is shown in Figure 3.

We also performed some experiments on a set of out-door images provided by the ENEA center of Casaccia (Rome).

The training set was made of four series of digital images shot by two close cameras along the four sides of an industrial building following the trajectory depicted in Fig. 4.

The aim of the robot guidance in this experiment is to identify the position of the robot along the trajectory by means of the 32 spots, where the pictures where shot, representing the control points for robot guidance. Each of these points was associated a numerical target obtained dividing the [0.3, 0.7] interval into 32 equi-length segments following the progressive ordering given. by the distance of the shot spot from a starting point; in other terms, the target represents the curvilinear coordinate of the robot on the trajectory.

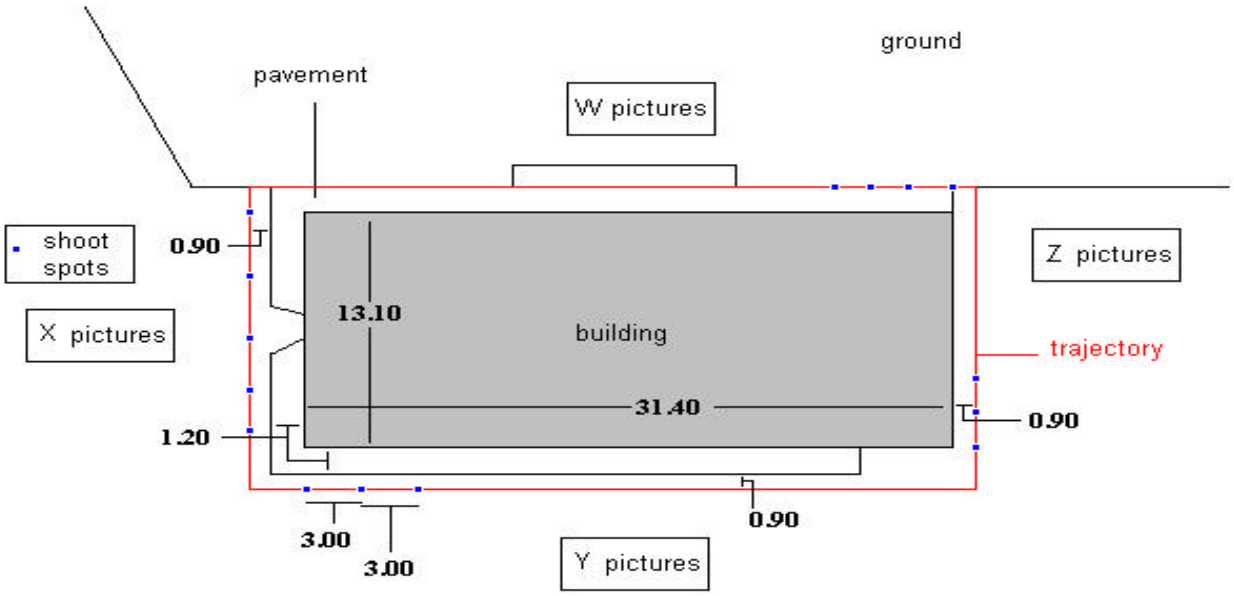

Figure 4. Map of the area where pictures for robot guidance were shot.

In this case we employed as initial configuration a 16:9:1 neural network trained on 10,000 learning epochs. The pruning procedure stopped with four kernels and a representation error of about 10^{-4}.

The centroids and the influence zones of the four RBFs are shown in Fig. 5.

Figure 5. Centroids and the influence zones in the out-door guidance problem (white circle: sampling areas after kernel adaptation; black circles: sampling areas after kernel adaptation).

5. PATH PLANNING AMONG UNKNOWN OBSTACLES

The robot is provided by distance sensors which detect the obstacles along suitable directions around it. The directions active during the two said ways of path planning implemented by the two algorithms are different. In the case of the NA, 54 directions are explored distributed as follows with reference to the direction of movement , which coincides with the perpendicular to the front of the robot:

- 21 directions uniformly distributed in the frontal region -30° ÷ 30° with steps of 3°. The angles are measured with respect to the said perpendicular;
- 15 directions uniformly distributed in the lateral regions 36° ÷ 120°, -36° ÷ -120° with steps of 6°;
- 3 directions located in the rear of the robot at 150°, -150°, 180°.

In the case of the SA, the sensors measure the distances in 12 directions uniformly distributed along the border of the robot every 30°.

The task pursued by the robot is to arrive to the target initiating its travel from a given point (the target). Since it is aware of its current location, it also "knows" the line which connects this location with the target (the "main line"). When all the obstacles in the environment are out of view, the robot moves along the main line. Only when some obstacle is detected by its distance sensors, the two said algorithms become active and determine the path.

The SA intervenes when the robot is "very close" to an obstacle. In order to understand how it operates, we note that the main line intersects the obstacle in an even number of points located on its border. Numbering these points with growing order toward the target, we note that there are always two paths along the border starting from point 1 and arriving to a successive "even" point (not necessarily that labeled with number 2). The SA guides the robot on a direction parallel to one of these two border paths and then leaves the obstacle continuing its movement toward the target along the main line. The choice of one of the two border paths is based on minimizing the angle between the current movement direction and that of the border path.

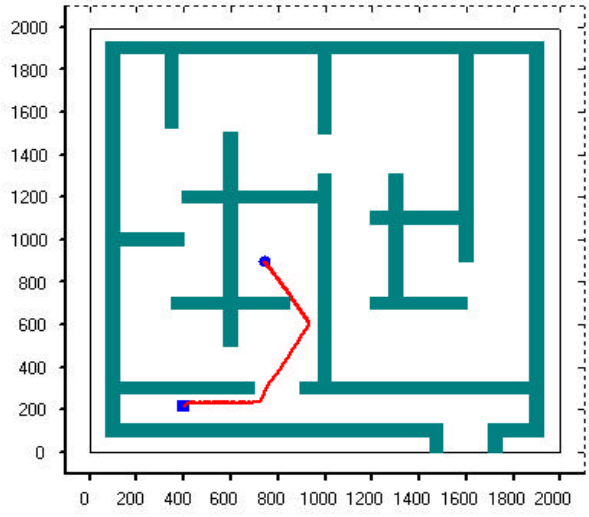

Fig 6: Application of the navigation algorithm to the environment 1.

There are also other alternatives (clockwise, counterclockwise, …); however the previous one was chosen since it yielded the best results. It is necessary to point out for clarity that it is always possible to follow one of the two border paths, since we assume the existence of a path from the starting point and the target. One of the border paths could be not feasible, if the obstacle under consideration is attached to the rectangle

delimiting the environment. If this is the case and the robot follows this unfeasible path, it will arrive to the side of the rectangle and then will be constrained to go back and initiate the other border path.

The NA is introduced for reducing the length of the path to the target. In fact, we note that the SA is able to guide the robot to the target, but it is not effective when the robot is far from the obstacles. The strategy it follows is similar to that of a person moving along a known direction in a dark environment in presence of unknown obstacles. He will avoid the obstacles by touching their borders in order to avoid them. In the case the environment is enlighten, the person will follow a completely different strategy.

It will look at the obstacles and then will plan a path between them in order of both avoiding them and reducing the length of the path. Such a strategy is the basis of the mechanism underlying the NA we have implemented.

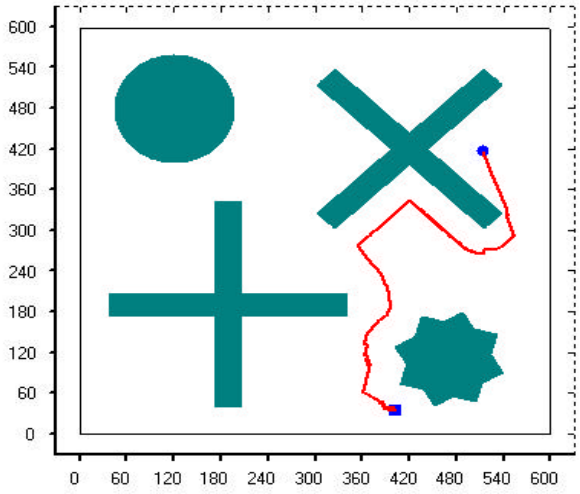

Fig. 7: Application of the navigation algorithm to the environment 2.

The SA acquires experience regarding the movement in presence of obstacles by means of a suitable neural network [Panella et al., 2001]. The network performs a clustering operation by modeling the density of the points representing the learning vectors with a mixture of Gaussian multidimensional functions determined via a maximum likelihood estimation. The learning vectors contain the 54 distances measured by the sensors in correspondence to particular locations of the robot in specific environments. The overall learning set is obtained by considering several environments and changing the types and locations of the obstacles and the start and target points. Since each vector corresponds to a known situation it is possible to optimize the successive movement of the robot by determining the angle between the main line and the optimized successive direction of movement (command-angle). This angle is associated to the learning vector and then by suitable averaging a command-angle is also associated with each cluster determined by the neural network.

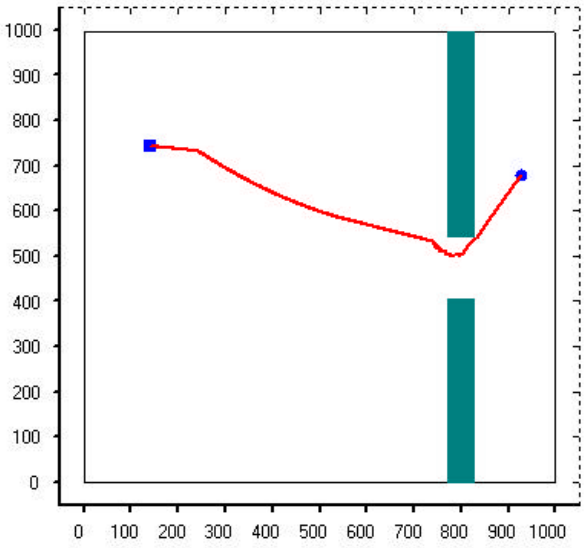

Fig. 8: Application of the navigation algorithm to the environment 3.

The effectiveness of NA depends on several factors:

- the experience gained by the robot is determined by the number and types of environments used in preparing the learning set. Its training can be pursued by applying a friendly package developed in the INFOM department of the University 'La Sapienza' of Roma, denoted as "Robot movements" and written in Microsoft Visual Basic 5. A short description of this software will be given in the appendix, while more details and the full user manual are available in [Restante, 1999];
- the command-angle determining the successive movement of the robot is associated with the distance vector either by the supervisor or through a heuristic procedure. Some improvements are however possible;
- the command-angle associated with a cluster is obtained by a simple averaging procedure. This process is the more sensitive factor of the

resulting NA. Because of the difficulty of improving it in the context of the clustering approach, we are also pursuing another approach based on function approximation. The command-angle is considered as a function of the distance vector determined in correspondence of the location of the robot.

In Fig.6-9 we show some simulation tests regarding the paths followed by the robot under the supervision of the two said algorithms in different environments. Some preliminary results concerning with this work were presented in [Martinelli et al., 1999] and several other results are available in [Restante, 1999].

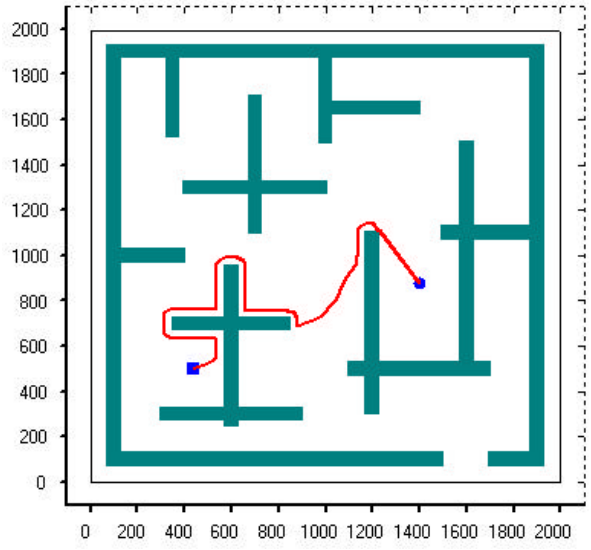

Fig. 9: Application of the navigation algorithm to the environment 4.

6. CONCLUSIONS

The present paper aimed at presenting a novel relevance determination technique for multi-layer perceptron networks and to verify its effectiveness on a engineering relevant task, such as mobile robot path following within an environment. The proposed technique allows network pruning by generalization maximization. In spite of the dramatic reduction of neural structure complexity, excellent numerical results were obtained both on test artificial problems and on real-world data, confirming the good performances of the proposed approach.

About obstacle avoidance, from the tests we carried out, it is possible to affirm that the proposed algorithm shows a good behavior in a wide set of environments. The use of neural networks for the NA allows to recognize “doors” between obstacles (see Fig. 8), and consequently the resulting planned path is much more efficient (in terms of its length) with respect to the plain SA algorithm.

REFERENCES

[Battiti, 1994] R. Battiti. Using Mutual Information for Selecting Features in Supervised Neural Net Learning. *IEEE Trans. on Neural Networks*, 3(4):537-550, 1994.

[Bishop, 1994] C.M. Bishop, Training with Noise is Equivalent to Tikhonov Regularization, *Neural Computation*, 7(1):108-116, 1995.

[Bollacker and Ghosh, 1996] K.D. Bollacker and J. Ghosh, Mutual Information Feature Extractors for Neural Classifiers. *Proc. of International Conference on Neural Networks (ICNN’96)*, pages 1528-1533, 1996.

[Burrascano, 1993] P. Burrascano, A Pruning Technique Maximizing Generalization, Proc. of International Joint Conference on Neural Networks (IJCNN’93), pages 347-350, 1993.

[Costa and Fiori, 2001] S. Costa and S. Fiori, Image Compression Using Principal Component Neural Networks, *Image and Vision Computing Journal* (special issue on "Artificial Neural Network for Image Analysis and Computer Vision"), 19(9-10):649-668, 2001.

[Dickmanns and Mysliwetz, 1992] E. Dickmanns and B. Mysliwetz. Recursive 3-D Road and Relative Ego-State Recognition, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14:199-213, 1992.

[Everitt, 1974] B. S. Everitt. *Cluster Analysis*. John Wiley & Sons Inc., 1974.

[Fiori and Burrascano, 2000] S. Fiori and P. Burrascano, Neural Network Feature Selection Applied to Robot Motion Control, *Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Complex Systems and Data Mining*, pages 77-82, 2000.

[Fiori et al., 2001] S. Fiori, F. Grimani, P. Burrascano, Novel Neural Network Feature Selection Procedure by Generalization Maximization with Application to Automatic Robot Guidance. Submitted to the *International Journal of Smart Engineering System Design*, 2001.

[Karnin, 1990] E.D. Karnin, A Simple Procedure for Pruning Back-Propagation Trained Neural Networks, *IEEE Trans. on Neural Networks*,

1(2):239-242, 1990.

[Kluge, 1993] K. Kluge, YARF: An Open Ended Framework for Robot Road Following, *Ph.D. Thesis, School of Computer Science*, Carnegie Mellon University, 1993.

[Kröse et al., 2001] B.J.A. Kröse, N. Vlassis, R. Bunschoten, Y. Motomura, A probabilistic model for appearance-based robot localization, *Image and Vision Computing Journal*, 19:381-391, 2001.

[Granvalet and Canu, 1995] Y. Granvalet and S. Canu, A Comment on Noise Injection into Inputs in Back-Propogation Learning, *IEEE Trans. on Systems, Man, and Cybernetics* 25(4):678-681, 1995.

[MacKay, 1992] D.J.C. MacKay, Bayesian Interpolation, *Neural Computation*, 4:415-447, 1992.

[Martinelli et al., 1999] G.Martinelli, F.M. Frattale-Mascioli, A. Rizzi, W. Iandolo, "Path planning by unsupervised neural nets for a planar robot navigating among unknown obstacles", European Conference on Circuit Theory and Design, pages 491-494, 1999.

[Lumesky and Stepanov, 1987] V. Lumesky, A. Stepanov. Path planning strategies for a point mobile automation moving admist unknown obstacles of arbitrary shape. *Algorithmica*, 2:403-430, 1987.

[Panella et al., 2001] M. Panella, A. Rizzi, and F.M. Frattale-Mascioli, A constructive EM approach to density estimation for learning, Proc. of International Joint Conference on Neural Networks (IJCNN'01), pages 2608-2613, 2001.

[Payton et al., 1993] D. Payton, J. Rosemblatt, and D. Keirsey, Plan Guided Reaction, *IEEE Trans. on Systems, Man, and Cybernetics*, 20(6):1370-1382, 1993.

[Restante, 2000] G. Restante, Uso di reti neurali per la navigazione di un ribot in un ambiente sconosciuto", Engineering thesis at the Engineering Faculty of the University 'La Sapienza' of Rome (in Italian), 1999.

[Setiono and Liu, 1997] R. Setiono and H. Liu, Neural-Network Feature Selection, *IEEE Trans. on Neural Networks*, 8(3):654-661, 1997.

[Thodberg, 1991] H.H. Thodberg, Improving Generalization of Neural Networks through Pruning, *Int. Journal of Neural Systems*, 1(4):317-325, 1991.

[Tishby et al., 1990] N. Tishby, E. Levin, and S.A. Solla, A Statistical Approach to Learning and Generalization in Layered Neural Networks, Proc. of IEEE, 78(10): 1568-1574, 1990.

[Yap, 1987] C. Yap. Algorithmic motion planning. In *Advances in Robotics, I: Algorithmic and Geometric Issue*. J. Schwartz and C. Yap Eds. Erlbaum Associates, Hillsdale, NJ, 1987.

APPENDIX

In order to measure and compare the performances of the proposed path finding algorithm, we designed a friendly graphic user interface (GUI); Fig. 10 shows its main window.

This GUI is mainly characterized by the following features:

- ✓ it allows to create environments, used to perform a simulation or to extract a proper training set;
- ✓ it is possible to generate a training set using multiple environments;
- ✓ on the basis of a training set, it permits the generation of a neural network; the user interface allows to select the desired metric and optimization procedure before the training procedure starts;
- ✓ it is possible to easily modify the associations between clusters and command-angles;
- ✓ a proper form allows to visualize the network structure;
- ✓ when performing a simulation, the user can easily set the algorithm to be used, the sensors' range, the simulation environment, the starting and arrival points.

An environment is a rectangular room in which it is possible to place some obstacles.

When designing an environment, the user can set the size of the room and decide the number, position and orientation of obstacles. The last version of our software allows to consider both rectangular and ellipsoidal obstacles.

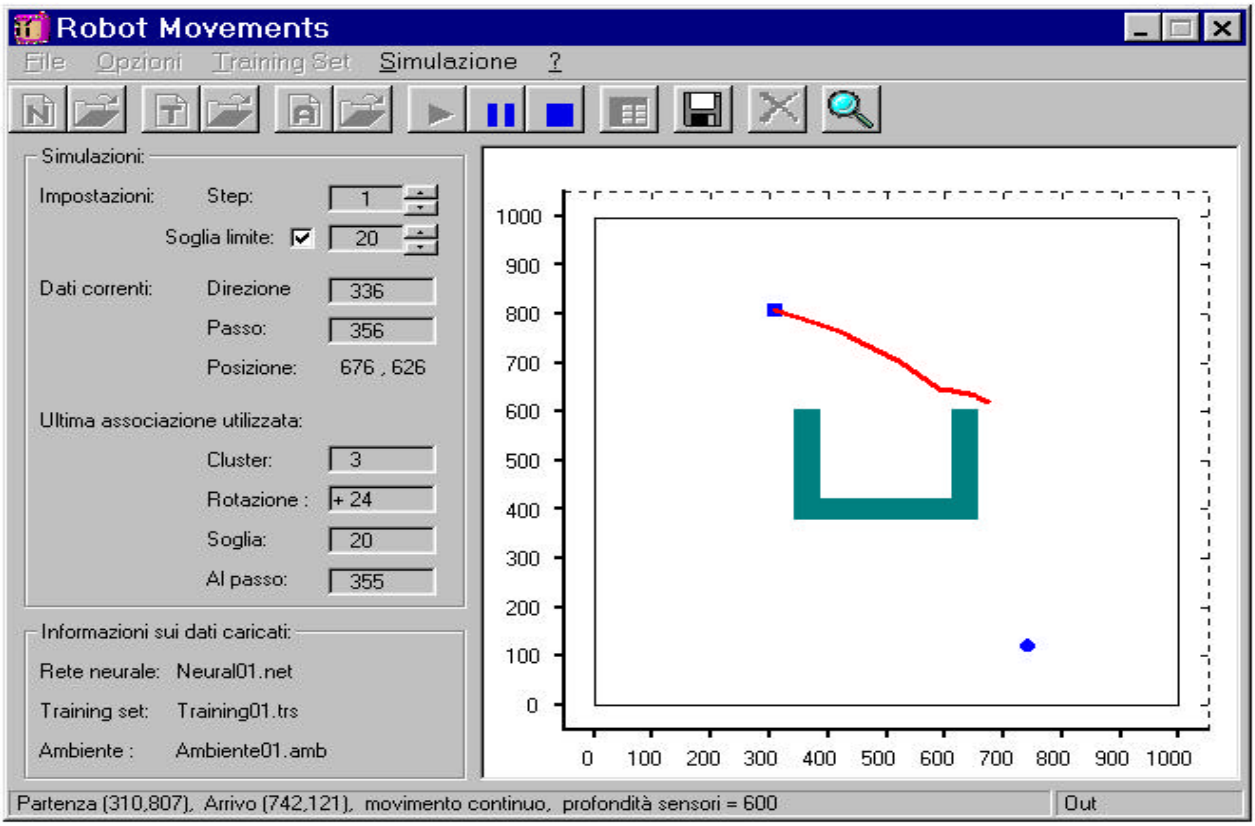

Fig. 10: The graphic user interface main form (in Italian).

ACKNOWLEDGMENTS

This research was carried out in co-operation with the ENEA Research Center (Casaccia-Rome) under grant PRASSI (``Massively parallel computational systems for real-time multiple-signal processing applied to autonomous robots for surveillance and security tasks", subgroup ``Neural-network-based algorithms for navigation and obstacle avoidance").