Case-Based Reasoning in Robot Indoor Navigation

Alessandro Micarelli, Stefano Panzieri, and Giuseppe Sansonetti

Department of Computer Science and Automation
Roma Tre University
Via della Vasca Navale, 79, 00146 Rome, Italy
{micarel,panzieri,gsansone}@dia.uniroma3.it

Abstract. In this paper, we advance a novel approach to the problem of autonomous robot navigation. The environment is a complex indoor scene with very little a priori knowledge, and the navigation task is expressed in terms of natural language directives referring to natural features of the environment itself. The system is able to analyze digital images obtained by applying a sensor fusion algorithm to ultrasonic sensor readings. Such images are classified in different categories using a case-based approach. The architecture we propose relies on fuzzy theory for the construction of digital images, and wavelet functions for their representation and analysis.

1 Introduction

Indoor robot navigation poses a unique challenge to Artificial Intelligence researchers. Mobile robots are inherently autonomous and they compel the researcher to tackle key issues such as uncertainty (in both sensing and action), reliability, and real time response. In particular, a still open problem is the devising of efficient strategies able to cope with the problem of *self-localization* in unstructured environments, i.e., the ability of estimating the position of the mobile platform when no artificial landmarks can be used to precisely indicate to the robot its position. To better explain this concept consider indoor navigation: the motion planning phase, that has to identify the best path through the environment must rely on the process of collection and interpretation of sensory data. This means that the robot, having no artificial landmarks, is asked to extract from natural features, like shape of corridors or lamps in the ceiling or even the number of encountered doors, the best estimate of its position. Moreover, the accuracy of such a process must be sufficient to plan its future actions. For this reason, to solve the self-localization problem for an autonomous mobile robot carrying out a navigation task consisting in moving between two points of a complex environment, the first step to take is characterizing an effective environment representation. This *map* must describe all the essential information being, at the same time, compact and easy to handle. Indeed, self-localization is always a multi-level process, usually consisting of more than one algorithm each

R.O. Weber and M.M. Richter (Eds.): ICCBR 2007, LNAI 4626, pp. 284–298, 2007.
© Springer-Verlag Berlin Heidelberg 2007

one related to the accuracy requested for the subsequent motion steps. When covering large distances, motion accuracy along the path is not demanding and the environment representation itself can be more rough. On the contrary, when approaching the goal, this ability must be improved to allow fine motion. So, assuming that the motion planning step has to be performed within this representation, the map must allow the description of the calculated path. This means that the efficiency of the map, in terms of flexibility, extendibility, and adaptability, must be considered as first goal in its design.

Now, suppose to restrict the problem and choose the environment in a particular class, still very wide: an office-like environment with corridors, corners and other similar features. Then, the task to perform can be described in linguistic terms containing topological elements such as "go straight along the corridor, turn right at first corner, and follow the next corridor as far as the second door on the left". Suppose also that only low cost sonar sensors can be used: all localization information, that at this point has a topological character, should be easily extracted from sensory data and used to guide the platform along the path. Unfortunately, in a dynamic environment, those features (*natural landmarks*) can vary and some unknown configurations could be found leaving to the robot the choice on several strategies: one could consist in finding the nearest matching topological element in a static library; an other one could include a supervised learning stage in which the new pattern is used to increase the base library itself. This second approach is often referred as Case-Based Reasoning (CBR) [1,7] and tries to catch all the learning opportunities offered both by the environment and, in an initial phase, by an external supervisor, to improve robot skill in analyzing its exteroceptive sensorial view.

The rest of this paper is organized as follows. In Section 2, we review different approaches to constructing a model of the environment based on sensor measurements. Section 3 presents the case-based architecture, in particular the signal representation and the similarity metric. Section 4 describes the experiments performed to evaluate the accuracy of the proposed system. Our final remarks are given in Section 5.

2 Mapping the World

In literature, the way the world is represented is found to be grouped into two main classes: *metric maps*, giving absolute geometric information about objects, and *topological maps* containing only relations between objects with no metric at all [2]. In general, topological maps can be more flexible due to their abstract world representation and can be successfully employed when there is no metric information or its quality is extremely poor. Moreover, a planar graph can be used to describe a topological map, and metric information, when present, can be introduced as weights on arcs or nodes.

Nevertheless, the semantic associated to nodes and arcs can differ depending on the authors. For example, in maps defined by [8] nodes represent *places* and are associated to sensory data and arcs represent *paths* between places and are characterized by control strategies. On the contrary, maps defined by [14] are obtained by analyzing probabilistic gridmaps (metric maps divided into small cells) and partitioning them into *regions* (nodes) separated by *narrow passages* (arcs). Finally, a different approach can be found in [6], where concepts from digital topology, extended to fuzzy gridmaps, are used to build a *topology-based map* in which structure and shape of the free-space is analyzed and classified: nodes of the graph represent connected components (usually rooms and corridors) and arcs represent adjacency relationships between these components. In this paper, we make use of a similar representation, that has been presented in [11], where connected components (nodes) are classified using a semantic induced by the particular shape, like corridors or corners, and arcs are again an adjacency relationship. The high level planner can force a navigation strategy associating to the particular node a behavior that the mobile robot must bind to while moving in that portion of the environment. This kind of autonomous navigation implies, therefore, a recognition phase for each step taken by the robot to estimate its position, or better, to understand the particular shape of the environment (the topological feature) inside its actual range of view.

In our case, this can be done comparing the actual sonar output with a set of reference signals associated with particular topological features. In most cases, association is done by comparing the actual view with a static list of models obtained with *a priori* considerations on the environment itself [5]. However, following a CBR philosophy, a learning approach can be devised in which real-world cases obtained from a supervised navigation are used to build and update a dynamic library.

In this paper, we want to show how such a method can be successfully applied to help the robot during navigation in dynamic environments containing features that only partially correspond to previously known cases. In particular, the problem we intend to address concerns the recognition of a sonar-based digital image and its classification under one category belonging to a set of predetermined topological situations (Corridor, Corner, Crossing, End Corridor, Open Space).

Basically, the surrounding of the robot is represented in terms of *Fuzzy Local Maps (FLM)*, i.e., *Fuzzy Maps* [10,11], that turned out extremely useful in many sensor fusion problems, obtained from a preprocessing stage applied to the sonar signals. Each FLM consists of 40 x 40 cells and, for each cell of an FLM, two values specifying the degree of membership to the set of empty cells and to the set of occupied cells are computed. An FLM, usually derived at each step merging the last n sets of collected data, is thereafter represented by two fuzzy sets: the empty cells set $\mathcal{E}$, and the occupied cells set $\mathcal{O}$. As an example, in Figure 1 the $\mathcal{E}$ set of a FLM obtained in a corridor is reported. Different gray levels in the image represent different fuzzy values. Pixels with darker gray levels correspond to lower values of membership to the empty cell set $\mathcal{E}$, white pixels are unexplored regions, with a fuzzy value of membership to $\mathcal{E}$ equal to zero.

Now, with reference to the scheme depicted in Figure 3, let us assume that the robot has acquired a new FLM. As first step, a feature-based representation of the new FLM is evaluated by the feature extraction module. This representation

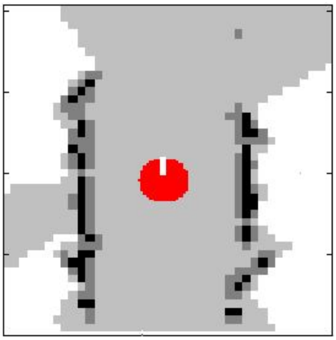

Fig. 1. Map of a corridor

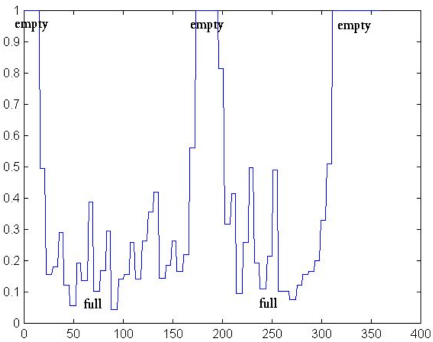

Fig. 2. Worldmark

constitutes the "new case" of the proposed CBR system. The retrieval module shown in the figure will effect a search in the Case Library containing the old cases, based on a <*problem representation*, *solution*> structure, which in this specific case will be <*FLM based representation*, *topological category index*>. The solution given in the old case can therefore be seen as a pointer to the "Library of Objects", containing the categories (i.e., "topological features") that could appear in the maps to be analyzed. The "recognized object" is at this point taken into consideration by the robot navigation system to plan its motion. This object, which constitutes the old solution of the case retrieved from the Case Library, will also be considered as a candidate solution of a new problem (basically, there is no need for an adaptation of the old solution to suit the new case) and if the human supervisor accepts it, the pair <*new FLM based feature representation*, *recognized object index*> can be inserted as a new case in the Case Library.

3 Case-Based Architecture

For sake of clarity and for an immediate understanding of the problems addressed and the relative proposed solutions, the pseudo-code of a rather simplified version of the classification algorithm is reported in Table 1. The complete solution, employed for the experimental performance assessment, was implemented in C language under the Linux operative system, for reasons of porting and efficiency. To handle both the new case and any of those cases dwelling in the Case Library, the use of a record structure comprising the three fields below was adopted:

- a one-dimensional fuzzy *worldmark* summarizing the content of the FLM;
- *object*, designed to store the label associated to the recognized object;
- *time*, reserved to the storage of information regarding the utility of the case of reference.

As indicated above, the first field is dedicated to the representation of the FLM. In order to guarantee the applicability of the current approach to real-time

Table 1. Pseudo-code for CBR

```
Function REC(NewImage) returns RecObject
inputs : NewImage; the input image
variables : CaseLib; the case library
            C_j; the generic old case
            T_nouse; the inactivity time
            S_a; the reliability threshold
            S_b; the identity threshold
local variables : D.image; the image representation
            D.object; the recognized object
            s_j; the metric value
            tempvalue; the temporary metric value
            tempind; the temporary case index

D.image ← WAVELET(NewImage)
D.object ← 0
tempvalue ← 0
tempind ← 0
for each old case C_j in CaseLib do
    begin
        s_j ← COMPARE_CASE(D.image, C_j.image )
        if (tempvalue < s_j) then
            begin
                tempvalue ← s_j
                tempind ← j
            end
    end
    if (tempvalue < S_a) then
        begin
            D.object ← HumanExpertSolution
            C_{n+1}.image ← D.image
            C_{n+1}.object ← D.object
            C_{n+1}.time ← 0
        end
    else
    begin
        if (C_tempind.object = HumanExpertSolution) then
                begin
                    D.object ← C_tempind.object
                    C_tempind.time ← 0
                end
            else
                D.object ← HumanExpertSolution
        if (tempvalue < S_b) then
            begin
                C_{n+1}.image ← D.image
                C_{n+1}.object ← D.object
                C_{n+1}.time ← 0
            end
    end
CLEAN_LIB(CaseLib,T_nouse)
RecObject ← D.object
returns RecObject
```

control, a simplification has been introduced: the bi-dimensional fuzzy map of Figure 1 is replaced with a one-dimensional fuzzy signal, named *worldmark*. The worldmark is computed by determining, for each direction around the robot, the value of the cell with the highest matching score to the set of empty cells, or, in other words, the cell for which the risk of belonging to a possible obstacle is minimum (see fig. 2). Therefore the “new case” that appears in Figure 2 consists of

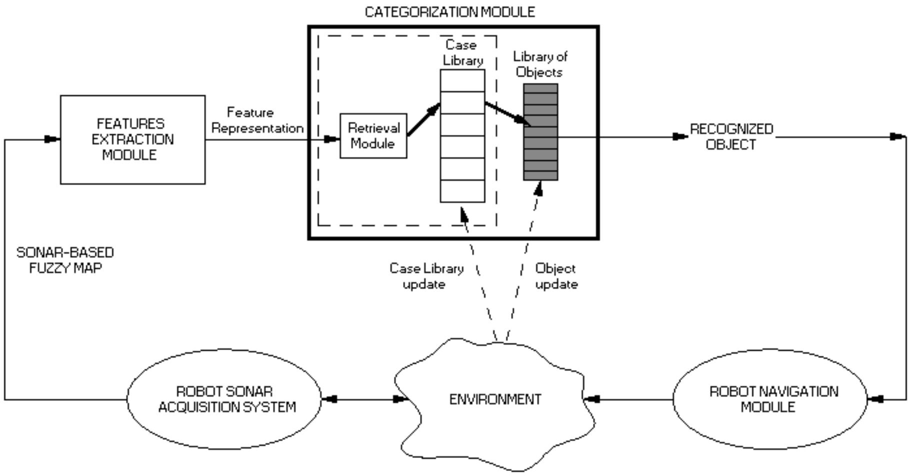

Fig. 3. Navigation architecture with Case-Based Reasoning

a vector of N elements (typically N=360) with values in the interval [0,1]. Before launching into the detailed description of the representation modalities of the aforementioned three fields, we believe it useful to provide a general overview of the entire algorithm. The domain expert's possibility to intervene in the decision task is certainly of primary interest. Such an intervention is possible both in the initial training phase of the system as well as during the verification phase for the retrieved solutions. The human element is, in fact, deemed indispensable not only when the robot begins to navigate without the support of any kind of information regarding the different topological configurations it may encounter, but also in the course of the regular operation of the system. In this way it is possible – *in fieri* – to remedy possible training shortcomings due to the limited information available. Another aspect worthy of attention is the one related to the adoption of a double similarity test. It is manifest that as the pertinence of the Case Library increases, so does the probability of retrieving a candidate with a good value of similarity to the case under examination and, therefore, that the associated solution to will prove to be valid even in a contingent situation. On the other hand, a rather voluminous library presents the two following inconveniences:

– more time necessary for the retrieval of the required information;
– a depletion in terms of available space.

It is evident that these problems relate across the board to any practical application of the CBR method. This becomes obvious when considering the considerable amount of work dedicated to the matter by the Artificial Intelligence community (see for instance [13]). In order to avoid, at least partially, this state of affairs, the proposed architecture uses two different tests, respectively, named *reliability test* and *identity test*. The former provides indications on the possibility of successfully apply the solution of the retrieved case to the new situation. It is a kind of measurement of the actual extent to which the case extracted from the library represents the class of the one under observation. Instead, the second test controls the insertion of the new case into the system memory. The reason for the introduction of the identity test parameter is owed to circumstances where it is useless to include a new case, "quite" similar to a case stored in the library in the system memory. In fact, such lesser information contribution would not justify the depletion of resources its storage would entail. The reliability test is performed by comparing the current similarity metric value s_j with the reliability threshold S_a, while the identity test is performed by comparing the same value s_j with an identity threshold S_b. In Tables 3, 4 and 5 the threshold values determined by a heuristic procedure are reported together with the percentage of coincidence between the responses given by the system and those provided by a domain expert. Specifically, for the setup of S_a and S_b, the available memory space, the amount of resources necessary to keep in memory the pair *<representation of signal, represented object>* and the statistics of the similarity index were considered. The results obtained by adopting such a strategy are more than satisfactory, but this does not deny the fact that an adaptive mechanism would certainly be preferable, i.e., one capable of dynamically determining the optimal values for the two thresholds on the basis of certain parameters established by the user, and in conformity with the structure of the overall system. This option is not yet a reality, but we believe that the resources necessary for developing such a solution could be actually quite contained. Keeping in mind an "intelligent" management of the resources available to the system, a third test has been introduced. The idea that has, concretely, lead to its introduction, stems from the need to keep track, for all cases stored in memory, of the *frequency* of their appearance and the *effectiveness* of the solution associated to them. The record field *time* was specifically introduced in consideration of these aims. Once more, the *clean library* test compares this value with a threshold T_{nouse}. If *time* exceeds T_{nouse} the case is removed from the library. For the determination of the optimal value to assign to the indicator T_{nouse}, the same considerations expressed above for the parameters S_a and S_b still apply. However, for a full understanding of the architecture proposed in this article there are still two major aspects that, as always, in any case-based system, constitute the heart around which all the rest revolves, that is,

- the signal representation;
- the similarity metric.

These aspects are, furthermore, strongly interrelated.

3.1 Signal Representation

Choosing the most efficient representation for a current problem constitutes the crucial moment of any application of signal processing. In fact, it is certain that the availability of a representation that makes the extraction of characteristics simple and immediate is of vital importance for the positive outcome of subsequent applications. Here, we resorted to a wavelet representation of the worldmark. The wavelet representation expresses the signal of interest as superimposed elementary waves and, therefore, in this respect does not introduce any innovation compared to traditional methods, such as Fourier series expansion. However, the innovative aspect offered by wavelet functions consists in the possibility of subdividing the available data in components with differing bandwidths and time durations. Each of these components is subsequently analyzed by a resolution associated to its scale. The advantages offered by this procedure are tangible, above all, in respect to the analysis of physical situations where typical signals show discontinuity and sudden peaks, exactly as happens with worldmarks. The advantages of adopting representations in similar situations through wavelet functions, instead of traditional methods, are extensively expounded in the literature [4,9,3].

The analysis procedure through wavelets is based on the use of a prototype function, called *mother wavelet*, whose translated and scaled versions constitute the basis functions of a series expansion which it is possible to represent the original signal with, by way of coefficients. Operations involving signals can, therefore, be developed – in a decidedly more straightforward and efficient way – directly on corresponding wavelet coefficients. If the choice of the mother wavelet is performed in an appropriate manner, i.e., if the coefficients below a certain threshold value are shrinked, it is possible to represent the original data *sparsely*, meaning with few coefficients different from zero. As a consequence, the wavelet constitute a formidable tool in the context of data compression and noise filtering in temporal series. Computation of the wavelet transform can be performed in a fast way (at a computational cost $O(n)$, where n is the number of signal samples) by means of the *Fast Wavelet Transform* (FWT) [9], a computationally efficient implementation of the Discrete Wavelet Transform (DWT) that exploits a surprising relationship between the coefficients of the DWT at adjacent scales. DWT can, moreover, be easily extended to multi-dimensional data, such as images, which may turn out to be useful in view of a possible application of our architecture to direct treatment of FLM, instead of the respective worldmarks. All these considerations induced us to rely heavily on the transformed wavelet in the context of our experiments.

3.2 Similarity Metric

The last aspect to be examined concerns the choice of the metric necessary for the evaluation of the *similarity* existing between case f in input and the generic case g belonging to the Case Library. The importance of this choice is due to its fundamental role in determining the quality of the selection procedure for the most promising case, which is the very essence of a CBR system. A review of any kind of CBR application will easily confirm this priority since the crucial role of the similarity metric selection is obvious in every instance.

Regardless of the application context, a good metric must anyhow be able to guarantee an efficient compromise between the two main requisites, which are the *quality* of the recognition and the *computational complexity*. Clearly, an evaluation procedure that offers excellent success rates but, at the same time, also requires excessive processing lengths, is not suitable for an autonomous mobile robot for which real time response is a mandatory requirement. On the other hand, it would be just as inexpedient to have a robot colliding, at high speed, against the first possible obstacle.

Accordingly, during the experimental activity several different metrics were tested, each of which revealed assets and shortfalls. Among them, the relatively best results were obtained by using the *cross-correlation factor* as metric, whose expression is:

$$\underset{\theta \in [0,2\pi]}{Max} \frac{\langle f(x), g(x-\theta) \rangle}{\sqrt{\langle f(x), f(x) \rangle \langle g(x), g(x) \rangle}}$$

In addition, when the worldmark is too noisy the similarity between the shrinked versions of the new and old worldmarks can be applied. This quantity was calculated both in the time and frequency domains, respectively, obtaining in both cases significant results with moderate processing time, through computation resources available on the market today.

4 Experimental Results

For our tests, we used the simulator of Nomad200 by Nomadic Technologies, a mobile robot equipped with a ring of 16 equally spaced ultrasonic sensors. The procedure consists of tracing a number of global maps of hypothetical office-like environments, simulating the robot dynamics and, finally, collecting the output data. For these operations we used the real time navigation software A.N.ARCH.I.C. [12] which, together with the aforementioned simulator, made the robot virtual navigation inside the mapped environment possible producing the sequence of FLMs and corresponding worldmark, each pair related to a different position taken during the followed path. Each sequence, therefore, includes hundreds of FLMs and worldmarks, which constitute the input for the tests that we performed on our classifier. The values reported below were obtained by using a machine equipped with a Pentium M processor, 1700 MHz, and 512 MB RAM. Before discussing the results obtained during the system test experiments, we deem it useful to set out here below some pertinent considerations. The recognition and classification of a digital image in one of the possible categories belonging to a predetermined set is a complicated task, not only for the machine itself, but also for humans. For example, imagine having to determine exactly the topological configuration that appears in the digital image shown in Figure 4. In this case, as may be easily discerned, it is not possible to affirm with absolute certainty that the robot is advancing along a corridor or is at a crossing, or an angle. This problem becomes increasingly complex as the robot approaches a transition situation between a perfectly defined configuration and its subsequent position. Similar considerations are clearly valid also for the corresponding worldmark, represented in Figure 5. Thus, one must generally bear these factors in mind during the analysis and evaluation phase of the results given by the experimentation. Indeed, the performance, as for any

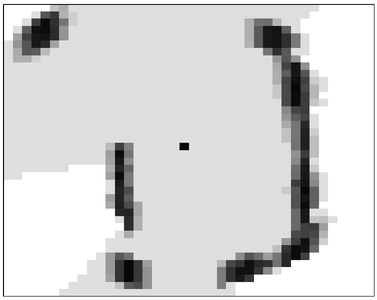

Fig. 4. Ambiguity map

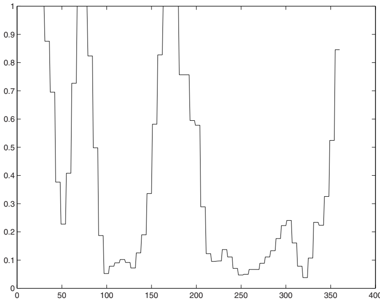

Fig. 5. Ambiguity worldmark

knowledge-based system, depends above all on the quality and quantity of the training effected before operating directly on the input data. Accordingly, during the testing phase, we initialized the system through representations related to four different configurations:

- corridor
- crossing
- end of corridor
- angle

providing, for each of these, three different standard schemes, in practice as it appears in the initial phase, at its basic level, and in the final phase. Table 2 shows the configuration of the system memory, at the time when the robot begins its navigation. We believe it necessary to stress how, in an architecture based on cases such as the one described here, the initial training constitutes only the first stage for the system acquisition of a knowledge base. Subsequently, during the normal course of recognition operations, the domain expert may intervene at any time if s/he deems it opportune, in order to enrich the Object Library. In practice, if it appears that the FLM does not represent any of the categories already in memory, following the similarity metric values adopted, nothing prevents the expert from envisaging and consequently introducing a new category. For example, if the robot goes against a well defined obstacle, say a desk, the human expert would have the possibility to intervene and assign the corresponding worldmark to a new class. All other worldmarks deemed similar to the one mentioned above, would subsequently belong, according to the used metric, to the new class. The ease and immediacy of such an operation constitute the strong points of the system presented herein. Tables 3, 4, and 5 show the results recorded during different series of tests of the system. Table 3 illustrates the results obtained by performing the similarity evaluation between the input signal and the generic one inside the Case Library directly in the time domain. Instead, for Tables 4 and 5, the same operation was effected in the wavelet domain, i.e., the matching evaluation of the two signals was not made by estimating the cross-correlation between sequences of temporal samples, but between the corresponding residual low-frequency components,

Table 2. Initial system memory

Number of cases represented	Case library	Object library
12	12	4

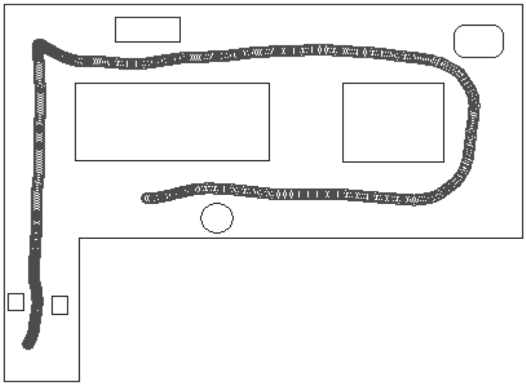

Fig. 6. Global map

obtained through DWT. Consequently, it is possible to appreciate in a more tangible way the extent of the possible advantages granted by the expansion of signals in series of waveform, perfectly located in time and in frequency. To perform this experimentation, we simulated the robot navigation in an environment that Figure 6 illustrates as a global map. In the same figure we have also traced the path followed by the robot, planned on the basis of specific methods for which further explanation is out of the scope of this paper. A sequence of 636 FLMs is thus generated, as well as a corresponding number of worldmarks. In order to streamline the experimental procedure, without, however, penalizing its efficiency, since the variation between one FLM and the subsequent one was practically insignificant, we decided to consider only one over three samples and to discard the others. As a result, the map effectively input to the system consists of only 212 FLMs. Initially, we shall examine the values reported in Table 3. As anticipated earlier, the tests were performed by running the system beforehand through the same training session, for each test series. This fact becomes apparent by looking at the data in the 6^{th} column, since the same value recurs systematically in each line (12 cases). As a matter of fact, the coincidence does not only concern the number of cases used, but also the samples themselves. In this way, we attempted to guarantee the same initial condition in each test series. A reading of the data discloses the consistency of the recorded fluctuations, in respect to the varying values assigned to the two similarity thresholds. For example, it is noticeable that when the reliability threshold S_a decreases, there is a proportional decrease in the number of interventions required of the domain expert by the system. Similarly, there is a clear increase in the number of cases

Table 3. Experimental results obtained in the time domain

S_a	S_b	Input Cases	Expert Interventions	Coincidence Percentage	Cases Before	Cases After	Processing Time (s)
0.88	0.91	212	12	90.1% (191)	12	37	8.07
0.88	0.93	212	12	85.4% (181)	12	51	9.97
0.88	0.95	212	11	84.4% (179)	12	74	13.55
0.89	0.91	212	14	90.1% (191)	12	37	8.17
0.89	0.93	212	15	85.4% (181)	12	51	10.07
0.89	0.95	212	14	84.9% (180)	12	74	13.71
0.90	0.91	212	15	90.1% (191)	12	37	8.32
0.90	0.93	212	15	85.4% (181)	12	51	10.04
0.90	0.95	212	14	84.9% (180)	12	74	13.66
0.91	0.93	212	23	91.9% (195)	12	51	10.35
0.91	0.95	212	19	89.6% (190)	12	74	13.89
0.93	0.95	212	30	90.6% (192)	12	74	14.30

inserted in the relative library matching an increase in the identity threshold S_b. However, the phenomenon of major importance and interest relates to the trend recorded by the factor indicated in the table as *coincidence percentage*. In previous sections of this paper, we dealt with the problem of finding a parameter that could, albeit roughly, provide an idea of the quality of the recognition and classification operations performed by the system. In accordance to such evaluations and also taking into account that it is necessary to assess the performance of a system that requires training, we deemed it expedient to adopt as evaluation factor the coincidence percentage gathered by a comparison between the system responses and those that would have been given by the same expert who performed the training, when examining the corresponding FLM. Clearly, such a strategy is inevitably damaged by the loss of information that occurs during the passage from a bi-dimensional fuzzy map (FLM) to the corresponding polar map (worldmark). However, notwithstanding this additional source of uncertainty, the results obtained may be considered more than satisfactory. Proceeding with the analysis of the data reported in Tables 4 and 5, which refer to the same experimental tests, but performed on the wavelet coefficients and not on their corresponding original signals, the gain is noteworthy, both in terms of coincidence percentage as well as computational complexity. In particular, it can be observed how the first factor is affected to a significant lesser degree by the variation of the values assigned to the two thresholds S_a and S_b. Although we do not wish to dwell upon too many details of the experimentation, it should be noted, however, that to obtain the wavelet coefficients relating to sequences of 360 temporal samples we applied a four-level DWT with analysis filters of the type of the Haar wavelet (Table 4) and the Daubechies wavelet with four coefficients (Table 5). The choice of these wavelets was due to the support size, that is, the size of the domain in which the wavelet function is nonzero. Selecting a wavelet with a small size of support is fundamental in order to characterize

Table 4. Experimental results obtained through DWT with the Haar wavelet

S_a	S_b	Input Cases	Expert Interventions	Coincidence Percentage	Cases Before	Cases After	Processing Time (s)
0.88	0.91	212	12	91.9% (195)	12	38	0.41
0.88	0.93	212	11	91.5% (194)	12	48	0.58
0.88	0.95	212	10	90.1% (191)	12	74	0.75
0.89	0.91	212	16	91.9% (195)	12	38	0.46
0.89	0.93	212	14	91.5% (194)	12	48	0.53
0.89	0.95	212	13	90.1% (191)	12	74	0.74
0.90	0.91	212	22	94.8% (201)	12	38	0.68
0.90	0.93	212	18	91.5% (194)	12	48	0.62
0.90	0.95	212	15	90.6% (192)	12	74	0.68
0.91	0.93	212	21	91.5% (194)	12	48	0.67
0.91	0.95	212	15	90.6% (192)	12	74	0.78
0.93	0.95	212	28	93.8% (199)	12	74	0.93

Table 5. Experimental results obtained through DWT with the Daubechies-4 wavelet

S_a	S_b	Input Cases	Expert Interventions	Coincidence Percentage	Cases Before	Cases After	Processing Time (s)
0.88	0.91	212	7	95.8% (203)	12	31	0.42
0.88	0.93	212	7	95.3% (202)	12	43	0.57
0.88	0.95	212	7	95.3% (202)	12	64	0.62
0.89	0.91	212	12	96.2% (204)	12	31	0.51
0.89	0.93	212	11	95.8% (203)	12	43	0.56
0.89	0.95	212	10	95.8% (203)	12	64	0.77
0.90	0.91	212	13	96.2% (204)	12	31	0.52
0.90	0.93	212	12	95.8% (203)	12	43	0.58
0.90	0.95	212	10	95.8% (203)	12	64	0.75
0.91	0.93	212	15	95.8% (203)	12	43	0.61
0.91	0.95	212	13	95.8% (203)	12	64	0.79
0.93	0.95	212	25	97.2% (206)	12	64	0.91

a signal with only few nonzero components in the transformed data. The Haar wavelet and the Daubechies-4 wavelet are the wavelets with the smallest support size [4]. This can be seen by the number of filter coefficients needed to represent each of them (two for Haar and four for Daubechies-4). Of the 360 coefficients of the complete DWT (it should be noted that being an orthogonal transformation there is coincidence between the number of signal samples to be transformed and the number of coefficients of the transformed signal) it was sufficient to only consider the 22 comprising the residual low-frequency component. The results show how the Daubechies-4 wavelet enabled us to better understand the dynamics of the input signal and to discard the phenomena ascribed to noise superimposed on the signal, which, on the contrary, considerably pollutes the values obtained when the entire original signal is analyzed. Another observation should be made on the processing time. In order to finalize this experimentation, for sake of clarity, we decided to operate on the group of worldmarks generated during the course of the overall navigation inside the simulated environment. Consequently, the time necessary to operate in real-time is decidedly less than that reported in the tables and, above all, significantly lower than the time allowed during the robot actual navigation.

5 Conclusions

Traditional methodologies of pattern recognition usually require the availability of templates of the objects we want to classify. This template collection reflects the a priori knowledge we have about the problem to be solved by the image classifier. However, in practical cases, as for the robot autonomous navigation, the prior knowledge could be rather poor, thus leading to a risk of misclassifications. In our contribution, we included a feature extraction algorithm into a CBR shell, which allows a constant update of the environment knowledge. We point out that, in principle, there is no limit to the number and complexity of information that may be collected in the Object Library, as well as in the Case Library. Future work will be focused on introducing the possibility of fusing more information coming from different kind of sensors (e.g., laser scanners or cameras) into a more detailed worldmark to supply the classifier with a better and more robust input data.

References

1. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications 7(1), 39–59 (1994)
2. Borenstein, J., Everett, H.R., Feng, L.: Navigating mobile robot: sensors and techniques. A.K. Peters, Ltd. Wellesley, MA (1996)
3. Chui, C.K.: An Introduction to Wavelets. Academic Press, London, England (1992)
4. Daubechies, I.: Orthonormal bases of compactly supported wavelets. Commun. Pure Appl. Math. 41(7), 909–996 (1988)
5. Fabrizi, E., Panzieri, S., Ulivi, G.: Extracting topological features of indoor environment from sonar-based fuzzy maps. In: Proc. of the 6th International Conference on Intelligent Autonomous Systems, Venice, Italy (2000)
6. Fabrizi, E., Saffiotti, A.: Extracting topology-based maps from gridmaps. In: Proc. of Int. Conf. on Robotics and Automation, San Francisco, CA (2000)
7. Kolodner, J.: Case-Based Reasoning. Morgan Kaufmann, San Mateo, CA (1993)
8. Kuipers, B., Byun, Y.T.: A robot exploration and mapping strategy based on a semantic hierarchy of spatial representation. Journal of Robotics and Autonomous Systems 8, 47–63 (1991)
9. Mallat, S.G.: A Theory for Multiresolution Signal Decomposition: the Wavelet Representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 11(7), 674–693 (1989)

10. Oriolo, G., Vendittelli, M., Ulivi, G.: Real-Time Map Building and navigation for Autonomous Robots in Unknown Environments. IEEE Transactions on Systems, Men and Cybernetics - Part B: Cybernetics 28(3), 316–333 (1998)
11. Panzieri, S., Petroselli, D., Ulivi, G.: Topological localization on indoor sonar-based fuzzy maps. In: Intelligent Autonomous System, pp. 596–603. IOS Press, Amsterdam, NL (2000)
12. Panzieri, S., Pascucci, F., Petroselli, D.: Auton. Navigation ARCHitecture for Intelligent Control (2002), www.dia.uniroma3.it/autom/labrob/anarchic
13. Schank, R.: Dynamic Memory: A Theory of Learning in Computers and People. Cambridge University Press, New York (1982)
14. Thrun, S.: Learning Metric-Topological Maps for Indoor Mobile Robot Navigation. Artificial Intelligence 99(1), 21–71 (1998)