

People Detection with Quantified Fuzzy Temporal Rules

Manuel Mucientes and Alberto Bugarín

Abstract—Detection of people and other moving objects is fundamental for the development of tasks by an autonomous mobile robot, and principally for human-robot interaction. In this paper we present an evolutionary algorithm to learn a pattern classifier system based on the Quantified Fuzzy Temporal Rules (QFTRs) model, for the detection of moving objects using laser range finders data. QFTRs are able to analyze the persistence of the fulfillment of a condition in a temporal reference by using fuzzy quantifiers. Experimental results with a *Pioneer II* robot in a typical hallway environment show an excellent classification rate in a real and complex situation with people moving in several groups in the surrounding.

I. INTRODUCTION

The presence of moving objects in the surrounding of the robot in a real environment is usual and must be taken into account. Typical environments are for example terminal areas of airports and railway stations, with people moving and carrying baggage (also moving objects), the hallway of a building, the corridors of a hospital, etc. In these environments the robot has to build maps, localize, plan routes, interact with humans or avoid obstacles. The knowledge of the position, speed and heading of the moving objects is fundamental for the execution of these and other tasks.

Several approaches to the detection of moving objects using laser range finders data have been proposed in the bibliography. We can group them in two categories: those that use the difference of occupancy between consecutive range scans [1], [2], [3], [4], and those that rely on specific characteristics of the moving objects, fundamentally people and, more specifically, legs of people [5], [6], [7], [8], [9].

All these approaches are based on heuristic rules and conditions. Thus, they work in a given environment and with an specific kind of moving objects but, if some conditions change, the method must be tuned. To solve this drawback, in this paper we present an evolutionary algorithm to learn a pattern classifier system based on Quantified Fuzzy Temporal Rules (QFTRs) for the detection of moving objects using laser range data. Our approach is based in the difference of occupancy between consecutive laser range scans.

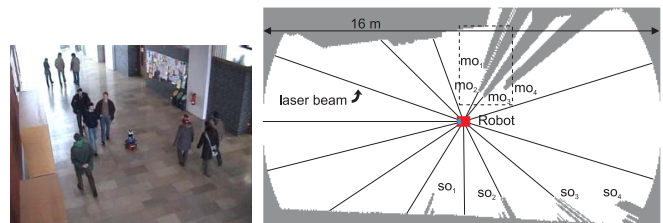
The novelties of our approach are two: first, the system is able to detect people that moves in groups (Fig. 1(a)). This usually happens in real environments, where people walk together, and carry objects (like suitcases, trolleys, etc.). In these situations, the density of moving objects is high, there are occlusions between moving objects, so the detection

and tracking difficulties are higher [10] than for individual objects. The second novelty is that the learned rules follow a paradigm, called QFTRs [11], that is able to represent and reason about values of variables evolving with time. Instead of classifying a pattern as a moving object by solely using the current values of some variables, QFTRs are able to quantify the fulfillment of a linguistic label by a set of data, and analyze the persistence of this fulfillment in a temporal reference.

The learned knowledge base has been tested with real data recorded with a *Pioneer II* robot equipped with two laser range scanners in a typical hallway environment. Results show an excellent classification rate over the test set. The paper is structured as follows: Sec. II shows the fundamentals of the detection of moving objects with laser range scanners. Sec. III presents the QFTRs model, while Sec. IV describes the evolutionary algorithm for the learning of QFTRs. Then, Sec. V shows the experimental results and, finally, Sec. VI points out the conclusions.

II. DETECTION OF MOVING OBJECTS

The detection of moving objects has been done using laser range finders data. These sensors emit several beams, each one in a direction. When a beam hits an obstacle, it is reflected and registered by the scanner's receiver. The time between the transmission and the reception of the pulse is known as the time of flight. With this information, the distance measured in the direction of each beam can be calculated. Fig. 1(b) shows a typical laser scan. The laser range finder provides the distances to the closest obstacle in each direction with a given resolution (number of degrees between two consecutive beams).



(a) Ten people moving in groups

(b) A typical laser scan

Fig. 1. Hallway environment

This work was supported in part by the Spanish Ministry of Education and Science under grant TIN2005-03844, and the DXID of the Xunta de Galicia under grant PGDIT04TIC206011PR.

Manuel Mucientes and Alberto Bugarín are with the Department of Electronics and Computer Science, University of Santiago de Compostela, E-15782 Spain (e-mails: {manuel, alberto}@dec.usc.es).

The existence of moving objects in the surrounding of the robot can be determined with two basic characteristics [1]:

- 1) A moving object appears in the distance histograms of the lasers as a local minimum.

- 2) As other static objects (like the legs of a table, a wastepaper basket, etc) can also have this characteristic, the object that has generated the local minimum must also be new at that position.

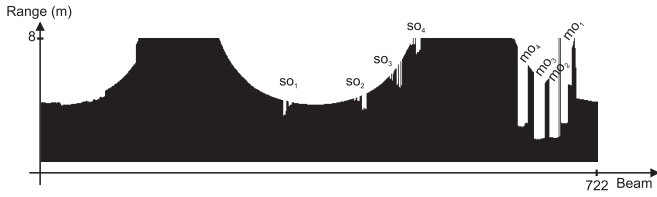


Fig. 2. Laser distance histogram corresponding to Fig. 1(b)

The first characteristic of a moving object can be observed in the laser distance histogram. Fig. 2 represents the distance measured for each beam of the laser for the laser scan in Fig. 1(b). Several local minima are detected. Those labeled as mo_m are moving objects, while the local minima identified by so_m are static objects. If we want to filter these static objects, we must identify which ones are new objects (they were not there in previous instants) and which were already there. This can be done constructing a local occupancy grid map of the environment.

Occupancy grid maps [12] represent the surrounding environment arranging it in cells of equal size (a grid). Each cell stores its probability of occupancy: a value of 1 indicates that the cell is occupied, 0 represents a free cell (without obstacles), while a value of 0.5 informs that the probability of occupancy is unknown (the cell has never been seen by the sensors, or it has been seen several times, sometimes occupied and others free). With each new laser scan the map is updated taking into account the current information provided by the sensors, but also the previous occupancy grid map.

Fig. 3(a) shows the occupancy grid map using the information of only one laser scan (rotated area inside the dashed rectangle of Fig. 1(b)). The map is represented in a 256 gray scale. The darker the gray, the higher the probability of occupancy of that cell. As this map represents only one laser scan, there are three possibilities: the cell is free (white), the cell is occupied (black), or the the probability of occupancy is unknown (gray). Fig. 3(b) represents the grid map in the same time instant, but this map contains now, not only the current laser scan, but also the scans of previous iterations. The positions of the moving objects, mo_m , have now a low probability of occupancy (light gray) as they have been detected as free in previous instants, and occupied only in the current scan. As we need to obtain the probability of new objects in the grid map, we will use the current map and the map t_b times before to calculate the probability of new objects for each cell:

$$P_{new}^{i,j}(t, t_b) = P_{occ}^{i,j}(t) \cdot (1 - P_{occ}^{i,j}(t - t_b)) \quad (1)$$

where $P_{occ}^{i,j}(t, t_b)$ is the probability of occupancy of cell with coordinates (i, j) using the sensorial information until the present instant (t) and $1 - P_{occ}^{i,j}(t - t_b)$ is the probability that

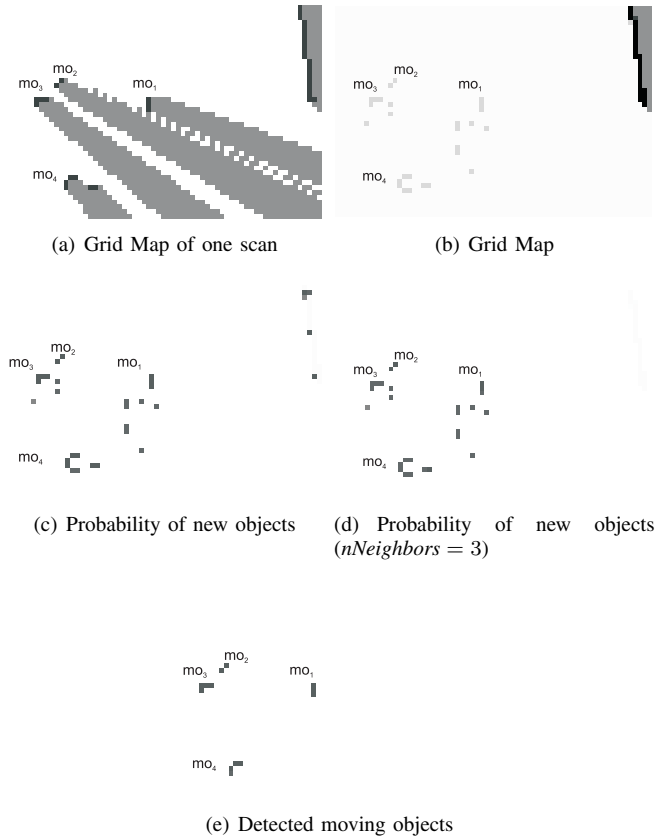


Fig. 3. Detection of four moving objects for the laser scan of Fig. 2

cell (i, j) is free in the map t_b times before. Fig. 3(c) shows the map obtained after applying Eq. 1 to the current (Fig. 3(b)) and previous grid maps ($t_b = 1$).

Due to small errors in the laser measurements and the size of the grid map cells, the same object can be detected in a cell in one laser scan and in an adjacent cell in other laser scan. These small errors heavily increase when the robot is moving, because of the odometric errors: the control commands of the robot are not precise, as the wheels can slip, turnings are not exactly implemented, etc. To eliminate some of these errors, a scan matching technique can be applied: we have tried the iterative point correspondence algorithm [13], but there are still errors that cannot be removed.

Therefore, the probability that an object detected in a cell is new needs to be calculated in a more reliable way. Errors can be filtered using a spatial window around each cell. Thus, Eq. 1 can be reformulated as:

$$P_{new}^{i,j}(t, t_b, nNeighbors) = \min_{k,l=-nNeighbors}^{nNeighbors} P_{occ}^{i,j}(t) \cdot (1 - P_{occ}^{i+k,j+l}(t - t_b)) \quad (2)$$

where $nNeighbors$ is the size of the window, and the probability that the object is new is calculated as the minimum over all cells of the window in the occupancy map t_b times before. The resulting grid map after applying Eq. 2 to the current (Fig. 3(b)) and previous grid maps is shown in Fig. 3(d) ($nNeighbors = 3$). As can be seen, the spatial window removes most of the points that belong to static objects. If

we combine the detected local minima with the probability of new objects (Fig. 3(d)) we obtain the detected moving objects (Fig. 3(e)).

The rules for the detection of moving objects must take into account information about the size of the local minimum (the gap between the object that produces the local minimum and the obstacles behind), the number of beams of the local minimum, and the probability of a new object in each of the cells of the local minimum. This last characteristic distinguishes between static and moving objects. The analysis of $P_{new}^{i,j}(t, t_b, nNeighbors)$ can also generate false positives. These errors can be reduced if instead of taking into account $P_{new}^{i,j}(t, t_b, nNeighbors)$ for an specific value of t_b , the system analyzes this probability in different time instants. Moreover, some cells of the local minimum can have a high probability of containing a new object, but others not. The system should also quantify how many cells must have a P_{new} over a threshold.

Therefore, a mechanism for performing the spatial filtering of successive values of $P_{new}^{i,j}(t, t_b, nNeighbors)$ is needed to produce a reliable detection of the moving objects. An example of a proposition that fulfills this objective is: “ P_{new} is high in most of the cells in part of the last instants”. We will implement such a proposition using the QFTRs model described in the following section.

III. QUANTIFIED FUZZY TEMPORAL RULES MODEL

The structure of a QFTR for the moving objects pattern recognition task is described in Fig. 4.

<p>IF X_{gap} is big_{gap} and (3) $X_{new}(nNeighbors)$ is $high_{new}$ in Q_s, new of the cells in Q_t, new of S_t, new and (4) X_{beams} is $high_{beams}$ (5) THEN <i>the pattern is a moving object</i></p>

Fig. 4. QFTR for the classification of a pattern as a moving object

Propositions (3) and (5) are non-temporal fuzzy propositions, while (4) is a Quantified Fuzzy Temporal Proposition (QFTP). Variable X_{gap} indicates the value of the gap (distance between the object that produces the local minimum and the obstacles behind). $X_{new}(nNeighbors)$ represents the probability of a new object in each cell of the local minimum and at each time instant for an specific value of $nNeighbors$ (Eq. 2): $X_{new}(nNeighbors) = \{x_{new}^{1,1}, \dots, x_{new}^{1,nCells}, \dots, x_{new}^{tBefore, nCells}\}$, where $tBefore$ is the number of analyzed time instants and $nCells$ is the number of cells of the local minimum. Finally, X_{beams} stores the number of beams of the local minimum.

The QFTRs model [11] is an extension of the FTRs model used in [14], [15]. QFTPs are of the form X is $A < in Q_s of > S_s < in Q_t of > S_t$, where X is a linguistic variable, A represents a linguistic value of X , Q_s and Q_t are fuzzy quantifiers, S_s is a fuzzy set, and S_t is a temporal reference

or entity. In proposition (4), S_s is not explicitly included as for all the values, $x_{new}^{p,w}, \mu_{S_s}(x_{new}^{p,w}) = 1$ (each value belongs with degree 1 to a cell).

The temporal entities S_t may represent both fuzzy temporal instants as well as fuzzy temporal intervals, being in both cases membership functions defined on a discrete set of values $\tau = \{\tau_0, \tau_1, \dots, \tau_p, \dots\}$, where each τ_p represents a precise temporal instant and τ_0 represents the origin. We assume that the values of this set are evenly spaced, where $\Delta = \tau_r - \tau_{r-1}$ is the unit of time, whose size or granularity depends on the temporal dynamics of the application. In our case, Δ is the time that passes between two consecutive laser scans. Temporal distributions are defined in relation to the current temporal point, τ_{now} (Fig. 5(a)).

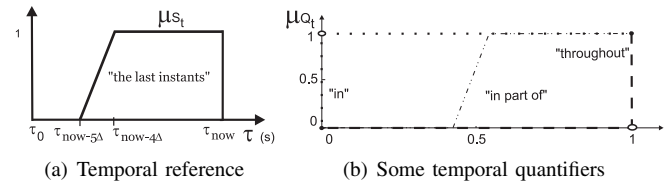


Fig. 5. Membership functions of a temporal reference and some quantifiers

The execution of a QFTR differs from that of a conventional fuzzy rule in the calculation of the Degree Of Fulfillment (DOF), which now also depends on the prior values of variables. A QFTP can be divided in two different parts:

- Non-temporal part: X is $A < in Q_s of > S_s$
- Temporal part: $\dots < in Q_t of > S_t$

The calculation of the DOF is implemented in the model in the following way: first, the degree of fulfillment of the non-temporal part of the proposition is calculated:

$$sc(\tau_p) = \mu_{Q_s} \left(\frac{\sum_{w=1}^{n\tau_p} \mu_A(X(\tau_p, w)) \wedge \mu_{S_s}(X(\tau_p, w))}{\sum_{w=1}^{n\tau_p} \mu_{S_s}(X(\tau_p, w))} \right) \quad (6)$$

where μ_A is the membership function of the linguistic label A , $X(\tau_p, w)$ is the value observed for variable X at the temporal point τ_p , and with non-temporal index w . w identifies the different values observed for variable X at a given temporal point τ_p , and takes values from 1 to $n\tau_p$. Finally, μ_{S_s} is the membership function of the fuzzy set S_s , μ_{Q_s} is the membership function associated to the linguistic quantifier Q_s , and the operator \wedge is the t-norm minimum.

Eq. 6 evaluates the percentage of points at τ_p that fulfill the linguistic label A and belong to fuzzy set S_s . This percentage of points is then evaluated with the linguistic quantifier Q_s , $\mu_{Q_s}(\dots)$, to obtain the non-temporal DOF of the proposition: $sc(\tau_p)$.

The second stage in the calculation of the DOF consists in the evaluation of the obtained $sc(\tau_p)$ within the temporal part of the proposition. The values of $sc(\tau_p)$ are modulated by the temporal part, so that the weight that is given to the temporal points is proportional to its membership in S_t , μ_{S_t} .

Different situations can occur for the fulfillment of the non-temporal part, X is $A < in Q_s of > S_s$, when S_t is an interval:

- Non persistence: as in “ P_{new} is high in most of the cells in the last instants”, where fulfillment is required for at least one point of S_t .
- Persistence: as in “ P_{new} is high in most of the cells throughout the last instants”, where fulfillment is required throughout the entire interval.
- Partial persistence: where the non-temporal part should be fulfilled for some subinterval (“in the majority of S_t ”, “in part of S_t ”), as in “ P_{new} is high in most of the cells in part of the last instants”.

The calculation of the DOF is implemented in a different way, depending on the type of temporal persistence of the QFTP:

- Non persistence: X is $A < in Q_s of > S_s$ in S_t

$$DOF = \bigvee_{\tau_p \in \tau} sc(\tau_p) \wedge \mu_{S_t}(\tau_p) \quad (7)$$

- Persistence: X is $A < in Q_s of > S_s$ throughout S_t

$$DOF = \bigwedge_{\tau_p \in \tau} sc(\tau_p) \vee (1 - \mu_{S_t}(\tau_p)) \quad (8)$$

- Partial persistence: X is $A < in Q_s of > S_s < in Q_t of > S_t$

$$DOF = \mu_{Q_t} \left(\frac{\sum_{\tau_p \in \tau} sc(\tau_p) \wedge \mu_{S_t}(\tau_p)}{\sum_{\tau_p \in \tau} \mu_{S_t}(\tau_p)} \right) \quad (9)$$

Operators \wedge and \vee are, respectively, the t-norm minimum and the t-conorm maximum, μ_{Q_t} is the membership function that is associated to the linguistic quantifier Q_t , and μ_{S_t} is the membership function of the temporal reference S_t . Fig. 5(b) shows some definitions for the membership functions (μ_{Q_t}) associated to the temporal persistence quantifiers that can be used.

The design of a QFTR involves the definition and tuning of ten parameters (linguistic labels and quantifiers) per rule for this application. Moreover, drastic changes in the characteristics of the environment or the moving objects could affect the accuracy of the pattern classifier system, making useless the tuned parameters. Therefore automated learning of Quantified Fuzzy Temporal Knowledge Bases (QFTKB) has to be used, in order to implement a new classifier for very different environmental conditions (corridor of a hospital, railway station, etc.). Also, in this way we can obtain a high classification rate, as the algorithm looks for an optimal solution. In the next section, an evolutionary algorithm to learn a pattern classifier system based on QFTRs for detecting moving objects is described.

IV. EVOLUTIONARY LEARNING OF QFTRs

Evolutionary learning of fuzzy knowledge bases has different approaches to represent the solution to the problem [16]. Our evolutionary algorithm follows the cooperative-competitive approach: each chromosome codifies a single

rule, and the solution to the problem is the complete set of individuals (rules). Rules evolve together (cooperative) in the population, competing among them to obtain a higher fitness. This approach needs to include a mechanism to maintain the diversity of the population (niche induction).

The coding scheme of the chromosomes of the population (Fig. 6) consists in three different parts, each of them corresponding to one of the propositions in a rule (Fig. 4). The linguistic labels and the quantifiers have been described with trapezoids, defined as a quadruple $\langle a, b, c, d \rangle$. Each gene (g) of the chromosome has an associated precision ($prec_g$) which codifies the value that indicates a meaningful change in a variable. For example, $prec_{new} = 0.01$, and $prec_{beams} = 1$.

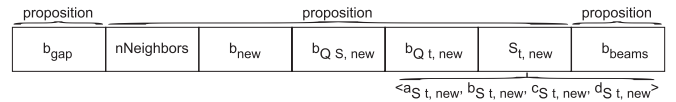


Fig. 6. Coding scheme of a chromosome

The first gene of the chromosome, b_{gap} , codifies the linguistic label big_{gap} (Fig. 4, proposition 3). The linguistic label is constructed as $\langle b_{gap} - prec_{gap}, b_{gap}, max_{gap}, max_{gap} \rangle$, where max_{gap} is the maximum value that variable X_{gap} can take. The same occurs with genes b_{new} ($high_{new}$), $b_{Q_s, new}$ (Q_s, new), $b_{Q_t, new}$ (Q_t, new), and b_{beams} ($high_{beams}$). On the other hand, $S_{t, new}$ is completely codified in the chromosome ($\langle a_{S_t, new}, b_{S_t, new}, c_{S_t, new}, d_{S_t, new} \rangle$). Finally, gene $nNeighbors$ indicates the size of the window used to calculate $P_{new}^{i,j}$ (Eq. 2).

The raw training and test data that have been used consist of a number of consecutive laser range scans (Fig. 1(b)). From these data, training and test examples sets are generated. The structure of an example e^l (moving object) is:

$$\{gap^l, nNeighbors^l, average_{pnew}^l, percentage_{cells}^l, t_b^l, beams^l\} \quad (10)$$

where $average_{pnew}^l$ is the average value of $P_{new}^{i,j}(t, t_b^l, nNeighbors^l)$ (Eq. 2) over all the cells occupied by the moving object. $percentage_{cells}^l$ is the percentage of these cells that have a value of $P_{new}^{i,j}(t, t_b^l, nNeighbors^l)$ (Eq. 2) over $average_{pnew}^l$.

It is relatively simple to label the moving objects from the laser scans to construct these examples sets, but this is not true for the static objects: for example, a wall can be a single static object, or due to its discontinuities could be also several objects. Due to this difficulty, our training and test examples sets will contain only moving objects. This does not represent a disadvantage, as the rules that are learned (Fig. 4) identify the class *moving object*, while any other pattern will be considered in the other class (*static object*).

A description of the evolutionary algorithm is shown in Fig. 7. First the population is initialized: a chromosome (Fig. 6) is generated for each example in the training set. Then all the individuals are evaluated. For each individual (rule) of the population, the following quantities are calculated:

- True positives (tp): sum of the DOFs of the examples that are moving objects and have been correctly classified by the rule.
- False positives (fp): sum of the DOFs of the patterns that are static objects and are incorrectly classified by the rule.
- False negatives (fn): these are the examples that are moving objects and have not been classified by the rule. $fn = n_{ex} - tp$, where n_{ex} is the total number of examples (remember that the examples sets only have examples of one class -moving object-).

True negatives are not defined as there are not examples of the class *static object* in the examples sets. False positives are defined as *DOFs of the patterns ...*, because there are not examples of the class *static object*, but there are static objects in the training and test data.

Taking into account the definitions for tp , fp , and fn , the accuracy of an individual of the population can be described as:

$$confidence = \frac{1}{1 + 10^p} \quad (11)$$

while the capacity of generalization of a rule is calculated as:

$$support = \frac{tp}{tp + fn} \quad (12)$$

Finally, we can define the $fitness_{raw}$ as the combination of both values:

$$fitness_{raw} = support \cdot confidence \quad (13)$$

which represents the strength of an individual without taking into account the others.

- 1) Initialize population
 - a) Generate rules
 - b) Evaluate population
 - c) Resize population (i)
- 2) for iteration = 1 to $maxiterations$
 - a) Crossover and mutation
 - b) Evaluate population
 - c) Resize population (ii)
- 3) Select rules for the final knowledge base

Fig. 7. Evolutionary algorithm

In the cooperative-competitive approach, a mechanism for niche induction must be included. The mechanism must promote the competition among individuals in the same niche (individuals that cover the same examples) while it must also preserve those individuals that have a low $fitness_{raw}$ if they are covering examples that are not covered by other individuals. Our algorithm uses the token competition [17] for this task: each example of the training set has a token and, of all the individuals that cover this example, the token will be seized by the individual with the highest $fitness_{raw}$. In this way, the individual with the highest strength in the niche will exploit it, while individuals that are weaker will reduce

its strength as they cannot compete with the best individual in the niche. Thus, the fitness of an individual is defined as:

$$fitness = fitness_{raw} \cdot \frac{seized_{ex}}{covered_{ex}} \quad (14)$$

where $seized_{ex}$ is the number of examples seized by the individual, while $covered_{ex}$ is the number of examples that have been covered by it (the DOF of the rule for the example is not null).

The final step in the initialization of the population is to resize it: all the individuals with null fitness are removed. This population is called *examples population*. From this *examples population* the first pop_{size} individuals are picked up to build the initial population. Finally with this initial population, the iterative part of the algorithm starts.

The first stage consists in the crossover and mutation of the individuals of the population. There is not selection. A couple of individuals is randomly picked up (all the individuals of the previous population have to be chosen once), the two individuals are crossed (with probability p_c), then mutated (with probability p_m), and finally added to the population. At the end of the process the population will double the size of the previous population, as it will contain the original individuals plus their offspring (due to crossover and mutation).

The crossover operator is the two-point crossover, while the mutation is implemented with two different operators: random mutation and step mutation. The algorithm selects between these operators, choosing random mutation with probability p_{rm} . Step mutation modifies the value of a gene (g) increasing or decreasing its value in a quantity of $prec_g$ and with equal probability. For the special case of gene $S_{t,new}$ that represents a trapezoid, the step mutation has three equiprobable options: shift, expand or contract the trapezoid.

After this stage, the population is evaluated (Eqs. 11, 12, 13, 14). Finally, population must be resized to pop_{size} : individuals with null fitness are removed and the best pop_{size} individuals are selected. Whenever the size of the population is under pop_{size} , new individuals are added. These individuals are chosen from the *examples population*, selecting those rules that cover examples that have not been seized yet by the individuals of the population. If the population is still under pop_{size} (this may occur in the last iterations of the algorithm), then mutated copies of the best individuals are inserted.

The last step of the algorithm is the selection of the rules. The final knowledge base will contain all those rules that have seized at least one example of the training set (non null fitness) and that fulfill that $seized_{ex} \geq fp$. This condition eliminates rules that generate false positives and seize a lower number of examples.

V. EXPERIMENTAL RESULTS

The training and test data sets have been obtained with a *Pioneer II* robot equipped with two laser range scanners. The lasers were mounted at a height of 40 cm (front laser) and 60 cm (rear laser), and with a resolution of 0.5 degrees. Thus,

one laser scan provides information of the whole surrounding of the robot. The experiment took place in the hall of a building (Fig. 1) of the University of Freiburg and lasted 9 minutes. The moving objects were in all the cases people moving in the hall, in groups of up to six people. Due to the disposition of the lasers, the legs of people have been detected.

The fact that people move in groups increases the difficulties in the detection, as compared with a single person, because cells that were originally free, are occupied in successive time instants by different moving objects (legs). Thus the values of $P_{new}^{i,j}(t, t_b, nNeighbors)$ are lower. During the experiment up to ten people moved around the robot, which means 20 moving objects at the same time. Such a high number of moving objects concentrated in a few and small areas of the environment generates partial (and total) occlusions of the objects, modifying the values of the gaps and the number of beams of the detected moving objects. Also, more than half of the time the robot was moving, making harder the discrimination of new objects as the scan matching errors increase.

We have extracted two time intervals from the data file to construct training and test examples:

- From time = 0 s to time = 45 s: the robot does not move, and there are 623 moving objects.
- From time = 101 s to time 123 s: the robot moves at 21 cm/s, and there are 609 moving objects.

The evolutionary algorithm has the following parameters: $maxIterations = 50$, $pop_{size} = 50$, $p_c = 0.9$, $p_m = 0.5$, $p_{rm} = 0.25$. Experiments have been performed with a 5-fold cross-validation: the examples set (1232 examples) was divided in five subsets of approximately equal size (around 246 examples). Then the learning process was run five times, using as training examples set four of the subsets, and testing with the remaining subset (this subset is different for each of the five runs).

TABLE I
RESULTS OF THE FIVE-FOLD CROSS-VALIDATION

	Rules	Training			Test		
		<i>fp</i>	<i>fn</i>	% correct	<i>fp</i>	<i>fn</i>	% correct
Average	25,60	11,40	19,60	96,85	2,20	7,40	96,10
σ	5,32	5,73	7,57	0,61	2,95	3,21	1,74

Results (Table I) show the average and standard deviation values of the five-fold cross-validation for the number of rules of the knowledge base, false positives (*fp*), false negatives (*fn*), and percentage of examples correctly classified (% correct) over the training and test examples sets. The percentage of examples correctly classified over the test set is 96,10%, although the number of moving objects, their high concentration in small areas of the hall, and the movement of the robot.

VI. CONCLUSIONS

We have developed a pattern classifier system based on QFTRs for the detection of moving objects using laser range

scan data. QFTRs have been learned with an evolutionary algorithm based on the cooperative-competitive approach together with token competition for niche induction. The system has been tested with real data obtained with a *Pioneer II* robot equipped with two laser range scanners. The moving objects are the legs of people (up to ten people at the same time) that were moving in groups in the environment. The experiment represents a real situation of a typical hallway environment. Both the number of moving objects and their high concentration in small areas, together with the movement of the robot, make the test conditions really complex. However, results show a very high average classification rate (96,10%) over the test set. In the near future we will run several tests in order to know the limitations of the system in relation with the number and density of people, as well as the analysis of its performance in a variety of completely different and real environments. Moreover, the pattern classifier system will be integrated with the algorithm proposed in [10] to track the detected moving objects.

REFERENCES

- [1] D. Schulz, W. Burgard, D. Fox, and A. Cremers, "People tracking with mobile robots using sample-based joint probabilistic data association filters." *Int. J. of Robotics Research*, vol. 22, no. 2, pp. 99–116, 2003.
- [2] M. Lindström and J.-O. Eklundh, "Detecting and tracking moving objects from a mobile platform using a laser range scanner," in *Proc. IROS*, Maui (USA), 2001, pp. 1364–1369.
- [3] C.-C. Wang and C. Thorpe, "Simultaneous localization and mapping with detection and tracking of moving objects," in *Proc. ICRA*, Washington DC (USA), 2002, pp. 2918–2924.
- [4] A. Fod, A. Howard, and M. Mataric, "A laser-based people tracker," in *Proc. ICRA*, Washington DC (USA), 2002, pp. 3024–3029.
- [5] N. Bellotto and H. Hu, "Multisensor integration for human-robot interaction," *The IEEE J. of Intell. Cybernetic Syst.*, vol. 1, 2005.
- [6] B. Kluge, C. Köhler, and E. Prassler, "Fast and robust tracking of multiple moving objects with a laser range finder," in *Proc. ICRA*, Seoul (Korea), 2001, pp. 1683–1688.
- [7] A. Brooks and S. Williams, "Tracking people with networks of heterogeneous sensors," in *Proc. ACRA*, Brisbane (Australia), 2003.
- [8] C. Martin, E. Schaffernicht, A. Scheidig, and H.-M. Gross, "Sensor fusion using a probabilistic aggregation scheme for people detection and tracking," in *Proc. ECMR*, Ancona (Italy), 2005, pp. 176–181.
- [9] G. Taylor and L. Kleeman, "A multiple hypothesis walking person tracker with switched dynamic model," in *Proc. ACRA*, Canberra (Australia), 2004.
- [10] M. Mucientes and W. Burgard, "Multiple hypothesis tracking of clusters of people," in *Proc. IROS*, Beijing (China), 2006, pp. 692–697.
- [11] P. Cariñena, A. Bugarín, M. Mucientes, F. Díaz-Hermida, and S. Barro, *Technologies for Constructing Intelligent Systems 2: Tools*. Physica-Verlag, 2002, ch. Fuzzy Temporal Rules: A Rule-based Approach for Fuzzy Temporal Knowledge Representation and Reasoning, pp. 237–250.
- [12] H. Moravec, "Sensor fusion in certainty grid for mobile robots," *AI Magazine*, vol. 9, no. 2, pp. 61–74, 1988.
- [13] F. Lu and E. Milius, "Robot pose estimation in unknown environments by matching 2d range scans," in *Proc. CVPR*, 1994, pp. 935–938.
- [14] M. Mucientes, R. Iglesias, C. V. Regueiro, A. Bugarín, P. Cariñena, and S. Barro, "Fuzzy temporal rules for mobile robot guidance in dynamic environments," *IEEE Trans. SMC-C*, vol. 31, no. 3, pp. 391–398, 2001.
- [15] M. Mucientes, R. Iglesias, C. V. Regueiro, A. Bugarín, and S. Barro, "A fuzzy temporal rule-based velocity controller for mobile robotics," *Fuzzy Sets and Systems*, vol. 134, pp. 83–99, 2003.
- [16] O. Cerdón, F. Herrera, F. Hoffmann, and L. Magdalena, *Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases*. World Scientific, 2001.
- [17] M. Wong, W. Lam, K. Leung, P. Ngan, and J. Cheng, "Discovering knowledge from medical databases using evolutionary algorithms," *IEEE Eng. in Med. and Biology Mag.*, vol. 19, no. 4, pp. 45–55, 2000.