Pattern Recognition 43 (2010) 1441–1453

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

People detection through quantified fuzzy temporal rules ☆

Manuel Mucientes *,[1], Alberto Bugarín

Department of Electronics and Computer Science, University of Santiago de Compostela, Spain

ARTICLE INFO

Article history:
Received 8 October 2008
Received in revised form
4 August 2009
Accepted 9 November 2009

Keywords:
People detection
Spatio-temporal pattern
Fuzzy temporal rules
Mobile robotics
Evolutionary algorithms

ABSTRACT

The knowledge about the position and movement of people is of great importance in mobile robotics for implementing tasks such as navigation, mapping, localization, or human–robot interaction. This knowledge enhances the robustness, reliability and performance of the robot control architecture. In this paper, a pattern classifier system for the detection of people using laser range finders data is presented. The approach is based on the quantified fuzzy temporal rules (QFTRs) knowledge representation and reasoning paradigm, that is able to analyze the spatio-temporal patterns that are associated to people. The pattern classifier system is a knowledge base made up of QFTRs that were learned with an evolutionary algorithm based on the cooperative-competitive approach together with token competition. A deep experimental study with a *Pioneer II* robot involving a five-fold cross-validation and several runs of the genetic algorithm has been done, showing a classification rate over 80%. Moreover, the characteristics of the tests represent complex and realistic conditions (people moving in groups, the robot moving in part of the experiments, and the existence of static and moving people).

1. Introduction

The operation of mobile robots in real environments, like supermarkets, railway stations, hospitals, etc., is generally characterized by the existence of people and moving objects in the surrounding. This fact needs to be considered when implementing tasks such as mapping or path planning, since discarding moving objects usually leads to errors and poor performance. The detection of people is particularly important for service robots and, fundamentally, for human–robot interaction, where both moving people and also static people have to be detected.

The detection of people is highly influenced by the type of sensor being used. The two types of sensors most widely employed for this purpose are cameras [1–4] and range finders (generally, laser range finders) [5–7]. The advantages of laser range finders are that they can directly measure objects geometry, distances information is accurate, the field of view is large, and information of the probability of occupancy of each area of the environment can be easily obtained. On the contrary, the quantity of information that can be extracted is lower than with a camera and, therefore, distinguishing among objects with similar geometric properties becomes much more difficult.

Several proposals have been done for the detection of people with laser range finders. They can be grouped into three categories: those that are based on the difference of occupancy between consecutive range scans [7–16] (motion-based), those that rely on geometric characteristics of people [5,6,17] (feature-based) and, finally, the approaches based on heuristic conditions, generally the width of the legs and the distance between them [18–22] (heuristic-based). On one hand, motion-based methods have two main drawbacks: they cannot usually detect people standing still, neither are able to distinguish between people and other moving objects (e.g., a trolley or a suitcase). On the other hand, heuristic and feature-based methods are not capable to differentiate between static and moving people.

One of the main difficulties for the detection of people for all the approaches is the presence of groups (clusters) of people, i.e., people that is walking or standing still together. This situation is very frequent in real environments. However, the higher the density of people, the tougher the detection. For motion-based approaches, whenever people is moving very close to each other, it is very probable that the position occupied by a person in a previous instant is now the current position of another person. Therefore, the difference of occupancy in that position would be the same as for a static object. Also, for feature and heuristic-based approaches, the high number of partial occlusions in the group of people changes the expected values of some of the features, decreasing the performance of the detection algorithm. Finally, another source of error in the classification of the detected

☆This work was supported by the Spanish Ministry of Science and Innovation under Grant TIN2008-00040.

* Corresponding author.
E-mail addresses: manuel.mucientes@usc.es (M. Mucientes), alberto.bugarin.diz@usc.es (A. Bugarín).

[1] The author is supported by the *Ramón y Cajal* program of the Spanish Ministry of Science and Innovation.

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.11.008

patterns for both approaches is the robot motion that produces an increase of the errors in the range data. This fact also enhances the difficulties for scan matching in the motion-based approaches.

In this paper, we present a pattern classifier system for the detection of people using laser range finders data. The idea underlying our proposal is a mixture of both the motion and feature-based approaches. The feature-based approach is able to detect spatial patterns, but if we want to independently classify static and moving people, the system should also analyze the difference of occupancy in consecutive scans. Moreover, our approach makes use of the fact that the analysis of a spatial pattern along time can improve the performance of the classifier. For these reasons, the proposed classifier system considers spatio-temporal patterns, i.e., patterns defined by the values of a set of features in specific time instants (spatial) together with their evolution along time (temporal). The classifier is based on a set of quantified fuzzy temporal rules (QFTRs), that are an extension of the fuzzy temporal rules (FTRs) model [23,24]. FTRs are able to represent and perform reasoning on values of variables evolving with time. Moreover, 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 combination of fuzzy logic with the learning capabilities of evolutionary algorithms is well known as evolutionary fuzzy systems. Evolutionary algorithms have some characteristics that make them specially adequate for learning fuzzy knowledge bases. The flexibility in the representation of the solutions is very high and, therefore, evolutionary algorithms can easily handle any type of fuzzy system. Also, depending on the characteristics of the problem and the demands of the final user, the designer has the possibility to determine the most appropriate trade-off between accuracy and interpretability by selecting different kinds of learning algorithms. In our proposal, a genetic algorithm has been used to learn the pattern classifier made up of QFTRs. It is based on the cooperative-competitive approach [25] together with token competition in order to maintain diversity in the population. After the learning stage, a selection process to find the best combination of rules from the obtained rule set has been applied using the Iterated Local Search (ILS) algorithm [26].

The most relevant novelties of our approach are:

- The learned rules follow the QFTRs paradigm, that is able to classify spatio-temporal patterns, i.e., patterns that require the analysis of the values of the features in specific time instants, but also the analysis of the evolution of these values along time intervals.
- The system is able to detect people that is moving in groups. In this situation, the density of moving objects is high and there are occlusions and partial occlusions between them, so the detection and tracking difficulties are higher [27] than for individual objects.
- The system is able to distinguish between static and moving people and, also, people can be detected even if the robot is moving and the errors in scan matching are high. The distinction between static and moving people provides the robot with more valuable information and improves the performance for tasks as tracking, path planning and human–robot interaction.

The paper is organized as follows: in the next section, we discuss related work. Section 3 shows the fundamentals of the detection of moving objects with laser range scanners. Section 4 presents the QFTRs model, while Section 5 describes the evolutionary algorithm for learning QFTRs. Then, Section 6 shows the experimental results and, finally, Section 7 points out the conclusions.

2. Related work

The algorithms described in the literature for the detection of people using laser range finders can be grouped into three categories: heuristic, motion and feature-based. Nearly all of them aim to detect the legs of people, although in [28] the algorithm detects the torso and the arms. Detection of people in heuristic-based approaches is done with a set of heuristic rules that try to detect meaningful characteristics of people patterns in the laser scan points. Many of the approaches that belong to this category rely on the detection of both legs of a person, and use the distance between them to filter false positives. However, when the environment is highly populated and, mainly when people walk in groups, detecting two legs of the same person is, unfortunately, not usual.

An example of an heuristic-based approach is presented in [18]. Legs are detected in three steps: remove noise, find the maximums and minimums of the distance function, and use some basic rules (basically, taking into account the inner distance between human legs) to discard false positives. The system was combined with a face detector to improve the performance. In [20], a leg is modeled as a semicircle with radius in a certain range, separated from its background by a threshold distance determined experimentally. A human is defined as a pair of legs within a certain distance of each other. Similarly, in [29] detection is based on grouping neighboring points into segments, classifying the segments as leg or non-leg according to a set of thresholds and, finally, grouping the detected legs into pairs depending on the distance among them. In [30], the detection system searches for objects of width similar to legs of people, and also considers the gap between a pair of legs. If only one leg is detected, the system requires further confirmation from a face detector. A similar idea is developed in [14]. The detection system accepts the following patterns: a single leg, two legs appropriately separated, and a person-wide blob. The last two patterns are always accepted as new person hypothesis, while the first one is only accepted when it is close to an already detected and tracked target.

Some of the heuristic-based proposals also use conditions closer to feature-based approaches. For example, in [31], the algorithm looks for leg-pairs. First, measurements are segmented into local groups. Then, each segment is checked for different conditions (width, deviation, etc.) and the distance between segments classified as legs is pairwise computed. In [19], objects are extracted from each laser scan following a two steps approach: scan segmentation and split of the segments into subsequences describing almost convex objects. To identify moving people, information about objects of successive scans is done computing the maximum flow with the minimal cost in a graph. Finally, in [28] the algorithm is able to detect human torsos and arms using raw sensor data. First, the range readings are thresholded using the background model (motion-based condition), and a set of candidate locations are created by identifying segments of contiguous foreground data points. The rules for this segmentation model approximate humans as ellipses, taking into consideration the possibility of partial occlusions and using heuristic rules for outliers rejection.

Motion-based approaches rely on the difference between consecutive scans, or between a scan and the background map of the environment. Their main lack is that these approaches cannot detect people standing still. A typical algorithm in this group is presented in [7,9]: it is based on the fact that people

walking result in local minima in the laser distance histograms. For each local minima, the system considers the changes in consecutive scans. In [10], they first construct the occupancy grid map of the environment, then calculate the difference of two consecutive maps and, finally, apply a blob segmentation algorithm to those cells occupied in the current map and free in the previous one. A similar approach is used in [32]; the moving object detection algorithm consists of two parts: to detect the moving points and to combine the results of segmentation and moving points detection for deciding which groups are potential moving objects. This decision is taken whenever the ratio of moving points in the group is greater than 0.5. In [11] moving targets were detected by frame differencing followed by erosion and connected components analysis. The system combined laser and panoramic vision. Also, in [33] two cues were used: body shape and motion. First they look for a single convex pattern of a certain size, representing the body of a person. After that, the system subtracts two consecutive scans.

In [13], detection is based on the construction of a closed polygon. The polygon is the free space detected by the scanner. Any point inside the polygon is considered as a moving object. In order to avoid false positives, the algorithm shrinks the polygons and applies a local minimization operator on the range readings across angles. In [8] moving objects are detected using a segmentation algorithm with three stages: identification of static obstacles (element by element comparison of closest points of consecutive scans), separation of moving and occluded obstacles and, finally, segmentation of the identified sets. In [12] the leg extraction algorithm, first, subtracts the background image, and then the moving points are temporally and spatially integrated. This stage is done accumulating the count of laser points at the same pixel in successive frames. After that, a Parzen window density estimation is applied and, finally, after a simple local search, the legs that remain static for a while in a small region are obtained. Finally, in [16] the detection system subtracts the background of each new scan, and those points that do not belong to the background are extracted and segmented into clusters. Next, the system looks for clusters that seem to be feet candidates (radius less than 30 cm). These clusters can be matched with the registered feet positions, or they can be paired if the distance is within a normal step size. Recognition of pedestrian is conducted by detecting the braided style of feet data: the algorithm calculates the distance between two clusters in the same group, and examines the sequence of distances to find if it appears as a periodic wave.

Feature-based approaches use geometric characteristics of the objects to be detected (legs of people in this case). For example, in [17] the leg detection system is based on circle detection, using a technique called inscribed angle variance, based on trigonometric properties of arcs. The detection of a leg uses the same prerequisites with the extra constraint of the distance between end-points falling within the range of expected leg diameters (0.1–0.25 m). A more elaborated approach was developed in [5]: a classifier to detect people was built using a supervised learning technique (AdaBoost). The algorithm used a set of geometric features from segments (number of points, width, linearity, radius, curvature, etc.) of groups of neighboring beams corresponding to legs of people. Based on the previous solution, in [6] the laser-based detection system consisted of two stages. The first stage corresponds to a classifier trained with AdaBoost algorithm (as described in [5]). The second stage uses the previous information of the position of legs to learn a probabilistic model that takes into account the possible distance between the persons legs and is more robust to occlusions and clutter.

Table 1 summarizes all these related papers. In most of the references, detection of people is not the main purpose of the paper, but a necessary stage prior to tracking or human–robot interaction. Moreover, in nearly all the papers, the values of the different parameters involved in the detection have been manually tuned. Only in Refs. [5,6] the classifier system was automatically learned (see "Learn" column of Table 1) and had detection of people as the primary objective.

Columns "Sensor mov.", "Clutt.", and "Populated" in Table 1 are related to the kind of tests performed to validate each proposal. In half of the papers, the laser sensors were moving (generally mounted on a robot) at least in one of the tests ("Sensor mov." in Table 1). This situation enhances the difficulties in the detection, as the robot motion increases the noise in sensor data. Another condition that makes classification harder is the existence of cluttered environments ("Clutt." in Table 1), as the

Table 1
Summary of related work.

Reference	*H.-b.*	*M.-b.*	*F.-b.*	Learn	Sensor mov.	Clutt.	Populated
Amarasinghe et al. [8]		×					
Arras et al. [5]			×	×		×	
Bellotto and Hu [18]	×				×		
Bennewitz et al. [9]		×			×		
Bobruk and Austin [10]		×					
Brooks and Williams [20]	×						
Chakravarty and Jarvis [11]		×					
Cui et al. [12]		×					×
Glas et al. [28]	×	×					
Kleinehagenbrock et al. [29]	×				×		
Kluge et al. [19]	×				×		×
Lindström and Eklundh [13]		×			×	×	
Martin et al. [31]	×					×	
Scheutz et al. [30]	×				×		
Schulz et al. [7]		×			×		
Topp et al. [33]		×					
Topp and Christensen [14]	×				×	×	
Wang et al. [32]		×			×		
Xavier et al. [17]			×				
Zhao et al. [16]		×			×		
Zivkovic and Kröse [6]			×	×	×	×	
QFTR-based proposal		×	×	×	×		×

H.-b., *M.-b.* and *F.-b.*, respectively, represent heuristic-based, motion-based and feature-based approaches.

legs of people can be confused with table legs, wastepaper baskets, etc. Finally, another difficulty is the presence of many people in the environment ("Populated" in Table 1), as the number of partial occlusions (modifying the characteristic leg patterns) is high. Also, for the motion-based approaches, the differences of consecutive scans do not always reflect the presence of a moving object, as a person can occupy the same position in which another person was previously placed.

The last row of Table 1 shows the characteristics of our proposal. The differences, from an experimental point of view, between our approach and the related work are:

- Only papers [5,6] presented a specific experimental study about the performance of the detection system. In the other papers, the quality of the classifier was only analyzed from a qualitative point of view, but no numerical results were shown.
- Our proposal works in populated environments and with the motion of the sensors. This combination is quite more complex to deal with than all the situations presented in previous approaches (in particular [5,6]), where the detection of people when the robot is moving in crowded environments is not considered.
- Our classifier distinguishes among three classes (static people, moving people, and static objects), while regarding [5,6] deal with two classes (people and objects).

3. Detection of people with laser range finders

Laser range finders emit at the same time beams in different directions. 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 flight time. With this information, the distance measured in the direction of each beam can be calculated. Fig. 1 shows a typical laser scan. The laser range finder provides the distances to the closest obstacle in each direction with a given angular resolution (number of degrees between two consecutive beams). Also, in this figure, some moving persons (mp_q) and static objects (so_q) with similar patterns to the persons have been labeled.

Our approach is a mixture of motion and feature-based approaches. A pure motion approach cannot be used, as we also want to detect static people. Also, the system must be able to distinguish between static and moving people and, therefore, motion-based conditions must be included. The first step in the detection process involves the segmentation of the current scan using the difference in the distance measured by consecutive beams. If that distance is over a threshold, a new segment is initialized. All these segments, $\Xi(t)=\{S_1^t,\ldots,S_{M_t}^t\}$, will be the input to the classifier system at time t. Each segment is a set of points in Cartesian coordinates: $S_m^t=\{\mathbf{s}_1^{t,m},\ldots,\mathbf{s}_{nPoints_m^t}^{t,m}\}$. The following features are used to classify people:

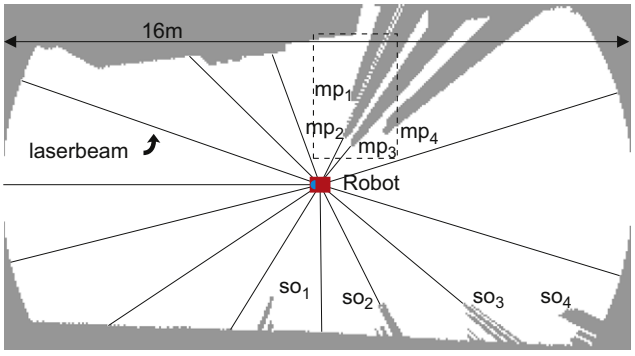

Fig. 1. A typical laser scan.

- The number of points of the segment, $nPoints_m^t=|S_m^t|$.
- The width of the segment, defined as the Euclidean distance between the first and last points of the segment: $width_m^t=\|\mathbf{s}_1^{t,m}-\mathbf{s}_{nPoints_m^t}^{t,m}\|$.
- The gap, which is the maximum of the differences of the norms[2] of the last point of S_{m-1}^t and the first point of S_m^t, and the last point of S_m^t and the first point of S_{m+1}^t: $gap_m^t=\max\{|\|\mathbf{s}_1^{t,m}\|-\|\mathbf{s}_{nPoints_{m-1}^t}^{t,m-1}\||,|\|\mathbf{s}_{nPoints_m^t}^{t,m}\|-\|\mathbf{s}_1^{t,m+1}\||\}$.
- The probability that the segment is new in its current position at time t, $P_{new}^{t,m}$. The estimation of this probability will be described in the remainder of this section.

From the human–robot interaction point of view, but also for mapping and path planning, it is interesting to distinguish between static and moving people. Therefore, a feature based on motion ($P_{new}^{t,m}$) was included as an input to the classifier. In order to discriminate between static and dynamic objects, it is necessary to compare the occupancy maps of the environment at the present and past time instants. Occupancy grid maps [34] represent the surrounding environment arranging it in cells of equal size (a grid). Each cell stores its probability of occupancy: 1 indicates that the cell is occupied, 0 represents a free cell (without objects), while intermediate values describe different degrees in the probability of occupancy. For example, 0.5 indicates an unknown occupancy (e.g., a cell that has never been detected by the sensors, or that 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. 2(a) shows the occupancy grid map using the information of a single laser scan (rotated area inside the dashed rectangle of Fig. 1). 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 probability of occupancy is unknown (gray). Fig. 2(b) represents the grid map in the same time point, but this map contains now both the current laser scan and also the scans of previous time instants. The positions of the moving persons, mp_q, 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 used the current map and the map at t' in order to calculate the probability of new objects for each cell:

$$P_{new}^{i,j}(t,t')=P_{occ}^{i,j}(t)\cdot(1-P_{occ}^{i,j}(t')) \quad (1)$$

where $P_{occ}^{i,j}(t)$ is the probability of occupancy of cell with coordinates (i,j) using the sensorial information until the current instant (t) and $1-P_{occ}^{i,j}(t')$ is the probability that cell (i,j) is free in the map at t'. Time instant t' is defined as $t'=t-\lambda\cdot\Delta$, where $\lambda\in\mathbb{N}^*$, and Δ is the elapsed time between two consecutive laser scans. Fig. 2(c) shows the map obtained after applying Eq. (1) to the current (Fig. 2(b)) and previous grid maps ($\lambda=1$).

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 another scan. These small

[2] Each vector is defined from the origin of coordinates to each point of the segment.

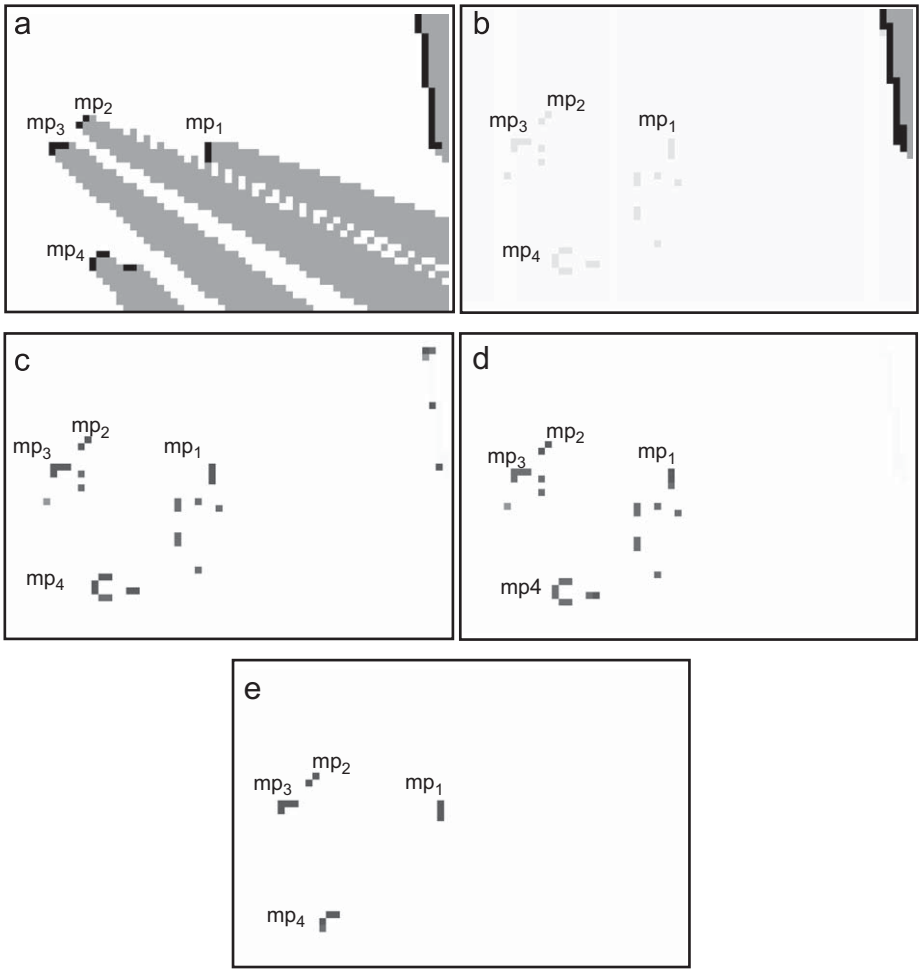

Fig. 2. Detection of four legs of people moving for the laser scan of Fig. 1. (a) Grid map of one scan. (b) Grid map. (c) Probability of new objects. (d) Probability of new objects ($size_w = 3$). (e) Detected people's legs.

errors highly increase when the robot is moving, because of the odometric errors: the control commands are not implemented with full precision, the wheels of the robot can slip, etc. The scan matching technique we have applied to eliminate some of these errors (iterative point correspondence algorithm [35]) was not fully able to remove all of them.

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

$$P^{i,j}_{new}(t,t',size_w) = \min^{size_w}_{k,l=-size_w} P^{i,j}_{occ}(t)\cdot(1\text{-}P^{i+k,j+l}_{occ}(t')) \quad (2)$$

where $size_w$ is half 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 at t'. The resulting grid map after applying Eq. (2) to the current (Fig. 2(d)) and previous grid maps is shown in Fig. 2(d) ($size_w = 3$). As can be seen, the spatial window removes most of the points that belong to static objects. Combining the probability of new objects (Fig. 2(d)) with the values of the features *gap*, *nPoints* and *width*, these segments can be classified as legs of moving people (Fig. 2(e)). Finally, the probability that a point of a segment is new can be obtained with

$$P^{t,m,size_w}_{new}(h,t') = P^{i,j}_{new}(t,t',size_w) : \mathbf{s}^{t,m}_h \in c_{i,j} \quad (3)$$

where $c_{i,j}$ is the cell with coordinates (i,j), and $\mathbf{s}^{t,m}_h$ is the h-th point of segment S^t_m.

The rules for the detection of moving objects must take into account information about the size of the gap between the analyzed segment and the adjacent ones, the number of points of the segment, the width of the segment, and the probability of a new object for each of the points of the segment. This last feature distinguishes between static and moving objects. The analysis of $P^{t,m,size_w}_{new}(h,t')$ can also generate false positives. These errors can be reduced if, instead of taking into account $P^{t,m,size_w}_{new}(h,t')$ for a specific value of t', 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. Thus, the system should quantify how many points must have a $P^{t,m,size_w}_{new}(h,t')$ over a threshold. The use of quantification introduces flexibility in the classification of the patterns. For example, a segment that has a few points with a low probability of being new could be classified as a moving object if the other points have a high probability of being new. In summary, a quantifier defines the percentage of points that must have a probability of being new over a threshold.

Therefore, the motion condition for the pattern classifier system, includes the probabilities of being new for the combinations of each point of the segment (h index) and each value for the time stamp of the previous grid (t'). Thus, a mechanism for performing the spatial filtering of the values of $P^{t,m,size_w}_{new}(h,t')$ for all the points is needed to produce a reliable detection of the moving objects. Moreover, the pattern classifier system also has to analyze the evolution of these values over time, in order to

implement a temporal filtering that can improve the performance and accuracy of the classifier. An example of a proposition that fulfills this objective is

$$P_{new}^{t,m,size_w}(h,t') \text{ is high in most of the points of the segment in part of the last instants} \quad (4)$$

We will implement such a proposition using the QFTRs model described in the following section.

4. Quantified fuzzy temporal rules model

The general structure of a QFTR for the people's legs pattern recognition task is described in Fig. 3.

Propositions (5)–(7) are non-temporal fuzzy propositions, while (8) is a quantified fuzzy temporal proposition (QFTP). This QFTP can be modeled according to the syntactic and semantic expressions described in [23,36,37]. In particular, proposition (8) is of the form

$$FS(t_p) \text{ in } Q_{\tau,new} \text{ of } F_{\tau,new} \quad (9)$$

where:

- $FS(t_p)$ is a filtered signal, i.e., a signal with a fuzzy value constraint. It is defined on the time axis and takes values in [0, 1].
- $Q_{\tau,new}$ is a fuzzy (temporal) quantifier ("part of" in expression (4)) defined in [0, 1].
- $F_{\tau,new}$ is a direct fuzzy temporal reference ("last instants" in expression (4)). It is defined on a discrete time axis $T=\{t_0,\ldots,t_{now}\}$, where each t_p represents a precise temporal point, t_0 represents the origin, and t_{now} the current time point. We assume that the values of this set are evenly spaced, where $\Delta = t_p\text{-}t_{p\text{-}1}$ is the unit of time, whose size or granularity depends on the temporal dynamics of the application.

Filtered signals are described in the model as signals whose values are constrained by a (spatial) reference or proposition. In this case, two constraints are involved:

$$FS(t_p) : P_{new}^{t,m,size_w}(h,t_p) \text{ is } F_{s,new} \text{ in } Q_{s,new} \text{ of } F_{segment}^{t,m} \quad (10)$$

where:

- $P_{new}^{t,m,size_w}(h,t_p)$ is the signal. In this particular case, it represents the probability that point h of segment S_m^t is new at time t_p.
- $F_{s,new}$ is an absolute fuzzy value constraint ("high" in expression (4)) defined in [0, 1].
- $Q_{s,new}$ of $F_{segment}^{t,m}$ is a quantified value constraint where:
  - $Q_{s,new}$ is a fuzzy (spatial) quantifier ("most" in expression (4)) defined in [0, 1].
  - $F_{segment}^{t,m}$ is a direct spatial reference ("points of the segment" in expression (4)).

IF gap_m^t is F_{gap} and (5)
$nPoints_m^t$ is $F_{nPoints}$ and (6)
$width_m^t$ is F_{width} and (7)
$P_{new}^{t,m,size_w}(h,t_p)$ is $F_{s,new}$ in $Q_{s,new}$ of $F_{segment}^{t,m}$ in $Q_{\tau,new}$ of $F_{\tau,new}$ (8)
THEN the pattern is a moving person

Fig. 3. QFTR for the classification of a pattern as a moving person.

Evaluation of this proposition produces a fuzzy filtered signal defined by

$$FS(t_p) = \mu_{Q_{s,new}}\left(\frac{\sum_{h=1}^{nPoints_m^t} \mu_{F_{s,new}}(P_{new}^{t,m,size_w}(h,t_p)) \wedge \mu_{F_{segment}^{t,m}}(h)}{\sum_{h=1}^{nPoints_m^t} \mu_{F_{segment}^{t,m}}(h)}\right) \quad (11)$$

using Zadeh's quantification model for proportional quantifiers ($Q_{s,new}$). This model evaluates the compatibility between the proportion of points that fulfill a given fuzzy property (in our case $P_{new}^{t,m,size_w}(h,t_p)$ being high and point h belonging to segment S_m^t) and the quantifier in the proposition ($Q_{s,new}$). $\wedge$ is the minimum t-norm.

The degree of fulfillment of the QFTP (expression (9)) is calculated as

$$DOF = \mu_{Q_{\tau,new}}\left(\frac{\sum_{t_p \in SUPP_{F_{\tau,new}}} FS(t_p) \wedge \mu_{F_{\tau,new}}(t_p)}{\sum_{t_p \in SUPP_{F_{\tau,new}}} \mu_{F_{\tau,new}}(t_p)}\right) \quad (12)$$

using again Zadeh's quantification model for proportional quantifiers ($Q_{\tau,new}$). Finally, expressions:

$$DOF = \bigvee_{t_p \in SUPP_{F_{\tau,new}}} FS(t_p) \wedge \mu_{F_{\tau,new}}(t_p) \quad (13)$$

$$DOF = \bigwedge_{t_p \in SUPP_{F_{\tau,new}}} FS(t_p) \vee (1\text{-}\mu_{F_{\tau,new}}(t_p)) \quad (14)$$

are used, respectively, for modeling existential ("in") and universal ("for all") quantifiers. These quantifiers correspond to the non-persistence and persistence situations for event $FS(t_p)$ in the interval $F_{\tau,new}$. This is a relevant characteristic of this temporal rule model, since it allows to consider partial, single or total fulfillment of an event within a temporal reference. $\wedge$ and $\vee$ are, respectively, the minimum t-norm and the maximum t-conorm.

4.1. An example of QFTP matching

In order to illustrate how a spatio-temporal pattern matches a QFTP, we will present an example. The definitions of the different membership functions and quantifiers are shown in Fig. 4. Moreover, $\mu_{F_{segment}^{t,m}}(h)=1$, $h=1,\ldots,nPoints_m^t$. This means that all the points are considered as parts of the segment, i.e., all the elements in the set have the same weight in the spatial pattern.

Fig. 5(a) shows the evolution along time of $P_{new}^{t,m,size_w}(h,t_p)$ for the five points that belong to the segment ($h=1,\ldots,5$). The values of the elements of the set, together with their evolution along time, conform a spatio-temporal pattern in the following way: for a given t_p, the set of values $\{P_{new}^{t,m,size_w}(1,t_p),\ldots,P_{new}^{t,m,size_w}(5,t_p)\}$ conform a spatial pattern. The evolution of this pattern along time produces the spatio-temporal pattern.

The first step for the classification of the pattern is the calculation of the degree of fulfillment of the fuzzy set $F_{s,new}$ for each value of $P_{new}^{t,m,size_w}(h,t_p)$ (Fig. 5(b)). Next, the filtered signal (Eq. (11)) has to be obtained. $FS(t_p)$ (Fig. 5(c)) represents the degree of fulfillment of the spatial pattern at each time instant. Finally, the degree of fulfillment (DOF) of the spatio-temporal pattern (Fig. 5(c)) is calculated following Eq. (12).

The design of a QFTR for this application (Fig. 3) involves the definition and tuning of four fuzzy linguistic labels, one temporal reference, and two fuzzy quantifiers, i.e., seven parameters per rule. Moreover, 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 learning of quantified fuzzy temporal knowledge bases (QFTKB) is of interest, in order to support the design/implementation of new classifiers for different applications (corridor of a hospital, railway

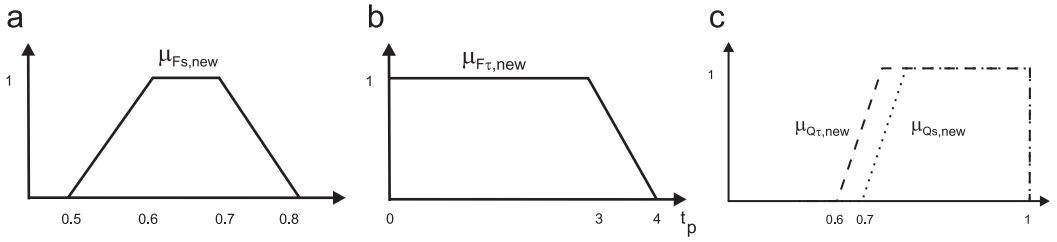

Fig. 4. Definition of membership functions and quantifiers for the QFTP example. (a) $F_{s,new}$ (e.g. "high"). (b) $F_{\tau,new}$ (e.g. "last instants"). (c) $Q_{s,new}$ (e.g. "most of") and $Q_{\tau,new}$ (e.g. "part of").

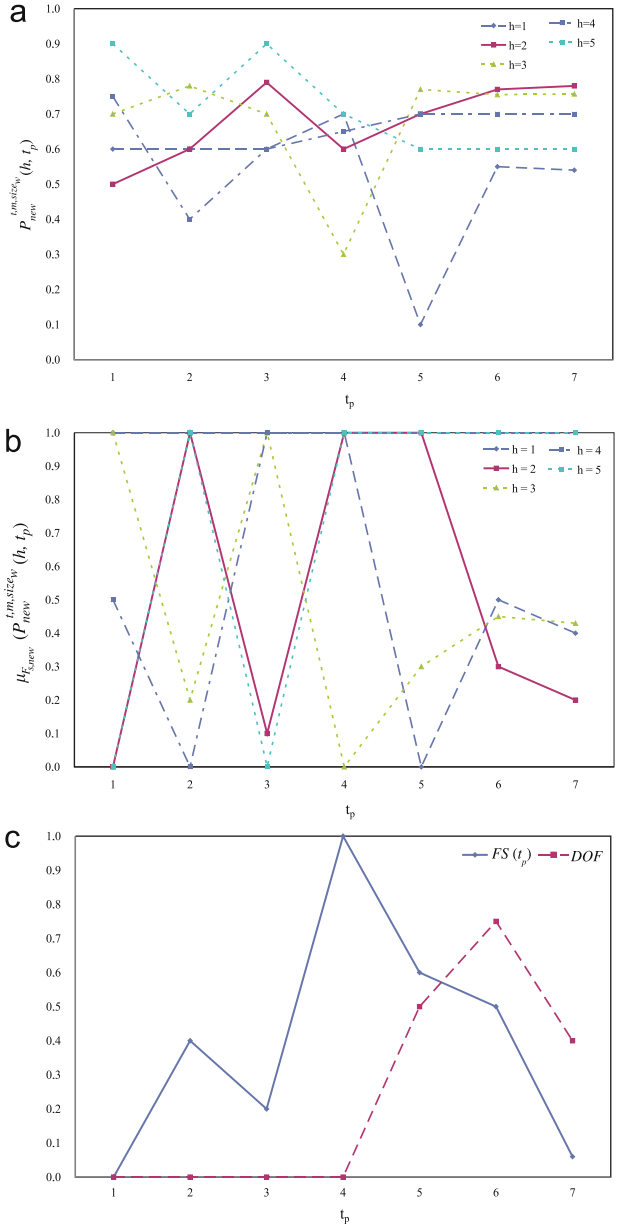

Fig. 5. An example of the evaluation of a QFTP for a spatio-temporal pattern. (a) Evolution along time of $P_{new}^{t,m,size_w}(h,t_p)$. (b) $\mu_{F_{s,new}}(P_{new}^{t,m,size_w}(h,t_p))$. (c) Filtered signal $FS(t_p)$ and DOF of the spatio-temporal pattern.

station, etc.). In the next section, a genetic algorithm to learn pattern classifier systems based on QFTRs for detecting people is described.

5. Evolutionary learning of QFTRs

According to [25,38], evolutionary learning of knowledge bases has different approaches to represent the solution to the problem: Pittsburgh [39], Michigan [40], iterative rule learning (IRL) [41], and genetic cooperative-competitive learning (GCCL) [42,43]. The proposed evolutionary algorithm follows the GCCL methodology. In this approach rules evolve together but competing among them to obtain the higher fitness. For this type of algorithms it is fundamental to include a mechanism to maintain the diversity of the population (niche induction). The mechanism must warrant that there is competition among individuals of the same niche, but also has to avoid the deletion of those weak individuals that occupy a niche not covered by other individuals of the population. We have chosen token competition [44,45] as the mechanism for maintaining the diversity. Its advantage over other approaches, like crowding or fitness sharing, is that it is not necessary to estimate the similarities between pairs of individuals.

The coding scheme of the chromosomes of the population (Fig. 6) consists of four different parts, each of them corresponding to one of the propositions of a rule (Fig. 3). Genes labeled as F_i are codified as trapezoids, $size_w$ is an integer, and the fuzzy quantifiers ($Q_{s,new}$, $Q_{\tau,new}$), although they are trapezoids, are also defined with one value, since the slope was kept fixed (Fig. 4(c)).

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

$$e^l = \{gap^l,\ size_w^l,\ average_{pnew}^l,\ Q_{s,new}^l,\ t_{pnew}^l,\ nPoints^l,\ width^l,\ C_l\} \tag{15}$$

where $average_{pnew}^l$ and t_{pnew}^l are defined as

$$average_{pnew}^l = \frac{\max_{t'} \sum_h P_{new}^{t,m,size_w^l}(h,t')}{nPoints^l} \tag{16}$$

$$t_{pnew}^l = argmax_{t'}\left(\sum_h P_{new}^{t,m,size_w^l}(h,t')\right) \tag{17}$$

and represent the maximum (over t') of the average probability of being new for the segment S_m^t, and its corresponding t' value. The generated examples sets have two different classes of examples: $C_l = mp$, which represents a moving person, and $C_l = sp$ which belongs to a static person. Finally, $Q_{s,new}^l$ (spatial quantifier) can be defined as

$$Q_{s,new}^l = \begin{cases} \dfrac{|\{h : P_{new}^{t,m,size_w^l}(h,t_{pnew}^l) \geq average_{pnew}^l\}|}{nPoints^l} & if\ C_l = mp \\ \dfrac{|\{h : P_{new}^{t,m,size_w^l}(h,t_{pnew}^l) \leq average_{pnew}^l\}|}{nPoints^l} & if\ C_l = sp \end{cases} \tag{18}$$

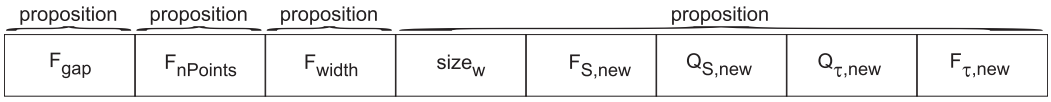

Fig. 6. Coding scheme of a chromosome.

representing the percentage of points of a segment, S_m^t, that have a probability of being new ($P_{new}^{t,m,size_w^l}(h, t_{pnew}^l)$) over ($C_l = mp$) or under ($C_l = sp$) the mean value ($average_{pnew}^l$).

It is relatively simple to label the static and moving people 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 also be several objects. Due to this difficulty, our training and test (example) sets will only contain static and moving people. This does not represent a disadvantage, as the rules that are learned (Fig. 3) identify the class *sp* or *mp*, while any other pattern will be considered in the default class (*static object*).

A description of the genetic algorithm is shown in Fig. 7. It is based on the GCCL approach with a population of variable size. The following sections describe each of the stages of the algorithm.

5.1. Initialization

The first step of the algorithm is the initialization of the population: a chromosome (Fig. 6) is generated for each example in the training set. Each trapezoid in the chromosome (Fig. 6) is initialized with a triangular membership function centered in the corresponding example value, and with the extremes of the triangle at a distance equal to $prec_i$ from the center of the triangle. $prec_i$ represents the lowest meaningful change in the value of variable i. On the other hand, the values for $size_w$ and $Q_{s,new}$ are directly copied from the example's values, and $Q_{\tau,new}$ is initially set to *non-persistence* (Eq. (13)). This population is called *examples population*. From this *examples population*, $pop_{maxSize}$ individuals are randomly picked up to build the initial population.

5.2. Selection

The first stage of the iterative part of the algorithm is the selection of the individuals of the population. We have tested the algorithm with two different selection mechanisms: binary tournament selection and uniform selection. On the one hand, in a k-tournament selection, k individuals are randomly picked from the population with replacement, and the best of them is selected. In this case, $k = 2$ (binary tournament selection). On the other hand, uniform selection means that each individual of the population is selected exactly once each time the selection process is performed.

5.3. Crossover and mutation

Once pop_{size} individuals have been selected, each couple of them is crossed with probability p_c. The crossover operator is the parent-centric BLX (PCBLX) [46,47]. Given two real-coded chromosomes, $X = (x_1 \dots x_g)$ and $Y = (y_1 \dots y_g)$ ($x_i, y_i \in [a_i, b_i]$, $i = 1, \dots, g$), that are going to be crossed, the following offspring are generated:

- $Z = (z_1 \dots z_g)$, where z_i is randomly selected from the interval $[l_i^z, r_i^z]$, with $l_i^z = \max\{a_i, x_i\text{-}I_i\}$, $r_i^z = \min\{b_i, x_i + I_i\}$, and $I_i = |x_i\text{-}y_i| \cdot \alpha$, $\alpha \in [0, 1]$.
- $V = (v_1 \dots v_g)$, where v_i is randomly selected from the interval $[l_i^v, r_i^v]$, with $l_i^v = \max\{a_i, y_i\text{-}I_i\}$ and $r_i^v = \min\{b_i, y_i + I_i\}$.

(1) Initialization
(2) for iteration = 1 to maxIterations
  (a) Selection
  (b) Crossover and mutation
  (c) Evaluation
  (d) Resize population
  (e) Check best population
(3) Rule subset selection

Fig. 7. Genetic algorithm.

This crossover operator is valid for genes representing real numbers. However, in the defined chromosomes (Fig. 6), there are also genes representing trapezoids. Given two trapezoids represented by tuples, $(\kappa_1 \cdots \kappa_4)$ and $(\rho_1 \cdots \rho_4)$ ($\kappa_j, \rho_j \in [a_j, b_j]$, $j = 1, \dots, 4$), that are going to be crossed, the following offspring are generated:

- $(\kappa_1' \dots \kappa_4')$, where κ_j' is randomly selected from the interval $[l_j^\kappa, r_j^\kappa]$, with $l_j^\kappa = \max\{a_j', \kappa_j\text{-}I_j\}$, $r_j^\kappa = \min\{b_j', \kappa_j + I_j\}$, $I_j = |\kappa_j\text{-}\rho_j| \cdot \alpha$, $\alpha \in [0, 1]$, $a_j' = a_j$ if $j \in \{1, 2\}$ or $a_j' = \kappa_2$ if $j \in \{3, 4\}$, and $b_j' = \kappa_3$ if $j \in \{1, 2\}$ or $b_j' = b_j$ if $j \in \{3, 4\}$.
- $(\rho_1' \dots \rho_4')$, where ρ_j' is randomly selected from the interval $[l_j^\rho, r_j^\rho]$, with $l_j^\rho = \max\{a_j', \rho_j\text{-}I_j\}$ and $r_j^\rho = \min\{b_j', \rho_j + I_j\}$. Finally, $a_j' = a_j$ if $j \in \{1, 2\}$ or $a_j' = \rho_2$ if $j \in \{3, 4\}$, and $b_j' = \rho_3$ if $j \in \{1, 2\}$ or $b_j' = b_j$ if $j \in \{3, 4\}$.

When crossover is not performed, both individuals are mutated with the non-uniform mutation operator [48]. If $X = (x_1 \dots x_g)$ ($x_i \in [a_i, b_i]$, $i = 1, \dots, g$) is a chromosome, and gene x_m is selected for mutation, then the resulting chromosome is $X' = (x_1 \dots x_m' \dots x_g)$ where

$$x_m' = \begin{cases} x_m + \Gamma(iteration,\ b_m\text{-}x_m) \text{ if } side = 0 \\ x_m\text{-}\Gamma(iteration,\ x_m\text{-}a_m) \text{ if } side = 1 \end{cases} \tag{19}$$

where $side \in [0, 1]$ is a random integer number, and function $\Gamma(iteration, val)$ is defined as

$$\Gamma(iteration, val) = val \cdot (1\text{-}r^{(1\text{-}iteration/maxIterations)^\upsilon}) \tag{20}$$

where $r \in [0, 1]$ is a random number, and υ is a parameter that determines the degree of dependency with the number of iterations. $\Gamma(iteration, val)$ generates a random value which has a higher probability of being close to 0 as the number of iterations increases. In this way, in the first iterations of the evolutionary algorithm the mutation operator explores the whole search space, while in the last stages it exploits the vicinity of the current value.

5.4. Evaluation

Before the evaluation of the population, the individuals of the former population are inserted in the current one. In this way, they will participate in the token competition. For each individual (rule) of the population, the following quantities are calculated:

- True positives:
  - $\#tp = |\{e^l : C_l = C_j \wedge DOF_i(e^l) > 0\}|$, where C_l is the class of example e^l, C_j is the class in the consequent of the i-th rule, and $DOF_i(e^l)$ is the *DOF* of the i-th rule for the example e^l.

#*tp* represents the number of examples that have been correctly classified by the rule.
  - $tpd = \sum_l DOF_i(e^l) : C_l = C_j$, i.e., the sum of the *DOF*s of the examples contributing to #*tp*.
  - $tp = \#tp + tpd/\#tp$.
- False positives:
  - $\#fp = |\{e^l : C_l \neq C_j \wedge DOF_i(e^l) > 0\}|$: number of patterns that have been classified by the rule but belong to a different class.
  - $fpd = \sum_l DOF_i(e^l) : C_l \neq C_j$, i.e., the sum of the *DOF*s of the patterns that contribute to #*fp*.
  - $fp = \#fp + fpd/\#fp$.
- False negatives:
  - $\#fn = n_{ex}^{C_j} - \#tp$, where $n_{ex}^{C_j} = |\{e^l : C_l = C_j\}|$. #*fn* is the number of examples that have not been classified by the rule but belong to the class associated to the consequent of the rule.

True negatives are not defined as there are not examples of the default class (static object) in the examples sets. False positives can be defined as there are static objects in the training and test data (laser scans), as well as elements of other classes. The values of *tp* and *fp* take into account not only the number of examples correctly/incorrectly classified, but also the degree of fulfillment of the rule for each of the examples. In case that $tpd \approx 0$, then $tp \approx \#tp$, while if it is high ($tpd \approx \#tp$) then $tp \approx \#tp + 1$. Taking into account these definitions, the accuracy of an individual of the population can be described as

$$\text{confidence} = \frac{1}{10^{fp}} \tag{21}$$

while the ability of generalization of a rule is calculated as

$$\text{support} = \frac{tp}{tp + fn} \tag{22}$$

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

$$\text{fitness}_{raw} = \text{confidence} \cdot \text{support} \tag{23}$$

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

In the GCCL 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 yet by other individuals. Our algorithm uses the token competition [44,45] 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

$$\text{fitness} = \text{fitness}_{raw} \cdot \frac{\text{seized}_{ex}}{\text{covered}_{ex}} \tag{24}$$

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 ($DOF_i(e^l) > 0$).

5.5. Resize population

Those individuals with null fitness are removed from the population, and the best $pop_{maxSize}$ individuals are selected for the final population. If $pop_{size} < pop_{maxSize}$, and there are still uncovered examples, then new individuals are added. These individuals are chosen from the *examples population*, randomly selecting those rules that cover examples that have not been seized yet by the individuals of the population.

5.6. Check best population

The output of the algorithm should be the best knowledge base throughout all the iterations. This could not be the population of the last iteration, as the algorithm is looking for the best single rules, but without taking into account the interaction among them. According to [49], a general model of fuzzy reasoning, which combines information provided by different rules, has the following steps:

(1) *DOF* or matching degree: represents the degree of activation of the antecedent part of the rule. For the *i*-th rule and pattern e^l, it is represented by $DOF_i(e^l)$.
(2) Association degree:

$$\phi_i^j = DOF_i(e^l)\delta_i \tag{25}$$

where δ_i is the weight or certainty degree of the classification of pattern e^l as class C_j (the one in the consequent of *i*-th rule).
(3) Pattern classification soundness degree for all classes. It is obtained with an aggregation function that combines, for each class, the association degree calculated for all the rules:

$$\psi_j = f(\phi_i^j, i = 1, \ldots, \#R) \tag{26}$$

where #*R* is the number of rules.
(4) Classification: the pattern is classified as class C_p, where $C_p = \max_j \psi_j$.

We have considered two different definitions for the weights (δ_i, Eq. (25)):

- No weight, which is represented as $\delta_i = 1, \forall i$.
- $\delta_i = \text{confidence}_i$, where confidence_i is the *confidence* (Eq. (21)) of the *i*-th individual (rule).

Moreover, a couple of aggregation functions (Eq. (26)) have also been analyzed:

- Maximum: $\psi_j = \max_i \phi_i^j$.
- Addition (also known as maximum vote): $\psi_j = \sum_i \phi_i^j$.

The performance of the knowledge base obtained after each iteration has been measured as

$$\text{fitness}_{pop} = 1 - \frac{\#fp + \#fn}{n_{ex}} \tag{27}$$

where n_{ex} is the total number of examples, and #*fp* and #*fn* are the number of false positives and false negatives obtained by the complete rule base, i.e., combining the information of all the rules that have been fired by the patterns that have to be classified. When the genetic algorithm ends, the population with the best fitness_{pop} is selected to build the pattern classifier system.

5.7. Rule subset selection

After the end of the iterative part of the algorithm, the performance of the obtained pattern classifier system can be improved selecting a subset of rules with better cooperation among them. This means that removing some of the rules of the final rule base, $fitness_{pop}$ can be increased. The rule selection

(1) Obtain the initial solution, Ω_0.
(2) $\Omega = localSearch(\Omega_0)$
(3) do
  (a) $\Omega' = mutate(\Omega)$
  (b) $\Omega'' = localSearch(\Omega')$
  (c) if $(fitness(\Omega'') > fitness(\Omega))$
    (i) $\Omega = \Omega''$
  (d) else
    (i) nRestarts++
(4) while $(nRestarts < maxRestarts)$
(5) Return Ω

Fig. 8. ILS algorithm [26].

process has the following steps:

1. Generate $\#R_{ga}$ rule bases, where $\#R_{ga}$ is the number of rules of the best population (RB_{ga}) obtained in the previous stage by the genetic algorithm. Each rule base is coded as: $RB_i = r_1^i \ldots r_{\#R_{ga}}^i$, with:
$$r_j^i = \begin{cases} 0 & \text{if } j > i \\ 1 & \text{if } j \leq i \end{cases} \quad (28)$$
where r_j^i indicates if the j-th rule of RB_{ga} is included ($r_j^i = 1$) or not ($r_j^i = 0$) in RB_i. With this codification, RB_i will contain the best i rules of RB_{ga}, as these rules have been ranked in decreasing order of their individual fitness (Eq. (24)). Notice that $RB_{\#R_{ga}}$ is RB_{ga}.
2. Evaluate all the rule bases, and select the best one, RB_{sel}.
3. Execute a local search on RB_{sel} to obtain the best rule set, RB_{best}.

The last step has been implemented with the iterated local search (ILS) algorithm [26]. ILS is based on the repetition of a local search using an initial solution that was obtained by mutation of a local optima previously found (Fig. 8). In this case, $\Omega_0 = RB_{sel}$, and localSearch() implements the local search of the best solution, i.e., it looks for the best neighbor and, if it is better than the current solution, it replaces the current solution and the local search is repeated again until no neighbors are better than the current solution. Also, for this implementation, the neighborhood of a solution, Ω, consists of all the solutions that have a Hamming distance to Ω under radius_{nbhood} (a parameter). Function mutate(Ω) creates a new solution that is a mutation of Ω, by randomly flipping the values at some positions. Finally, if the solution found in the current iteration (Ω'') is not better than the current solution (Ω), a new iteration is executed if the number of restarts (failures of the search) is under a threshold (maxRestarts).

6. Results

Two datasets have been used in order to validate both the pattern classifier system model (based on QFTRs) for the detection of people, together with the evolutionary algorithm to learn QFTRs for classification. The main characteristics of the datasets are shown in Table 2, where $time_{moving}$ is the time that the robot was moving in the environment, and $\max_t\#mp$ and $\max_t\#sp$ are the maximum number of legs of moving and static people in the environment at a time instant.

Dataset *uscCoie* was recorded in a university students information office (Fig. 9(b)), and contains both moving and static people. Two lasers were mounted on a platform at a height of approximately 40 cm, with a resolution of 0.5° and covering the whole surrounding. Due to the disposition of the lasers, the legs of people have been detected. The difficulty of the dataset is the distinction between patterns of both classes, also in an environment with a high number of people and where persons are close to each other.

Table 2
Characteristics of the datasets.

Dataset	n_{ex}	Time (s)	$time_{moving}$ (s)	$\max_t\#mp$	$\max_t\#sp$
uscCoie	1,115	13.67	–	6	14
uFrei	1,348	61.38	21.82	20	–

uFrei dataset was 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°. Thus, one laser scan provides information of the whole surrounding of the robot. The experiment took place in the hall of a building (Fig. 9(a)). The moving objects were in all the cases people moving in the hall, in groups of up to six people. Again, as the lasers were placed fairly low, the legs of people were detected. The fact that people move in groups increases the difficulties in the classification, changing the typical detection pattern of a single person. This is due to two main reasons: partial occlusions and lower values of $P_{new}^{t,m,size_w}(h,t')$ (Eq. (3)). Fig. 10 shows a typical example of partial occlusions with 10 legs of people (mp_q). The patterns corresponding to mp_5 and mp_6 are the typical patterns of one leg. However, the pattern corresponding to $mp_1, \ldots, mp_4$ shows a partial occlusion of mp_3, a total occlusion of mp_4, and two patterns (mp_1 and mp_2) that are very close. Thus, the characteristics of the patterns are different from the typical ones. A similar situation occurs with $mp_7 - mp_8$ and $mp_9 - mp_{10}$, that cannot be distinguished due to their proximity. In relation to $P_{new}^{t,m,size_w}(h,t')$, the presence of groups of people reduces that values because cells (in the grid map) that were originally free, are occupied in successive time instants by different moving objects (legs).

Tables 3 and 4 show the results for datasets *uscCoie* and *uFrei*. Each row in the tables represents the results of the evolutionary algorithm for a five-fold cross-validation and 10 runs per partition. A k-fold cross-validation consists in dividing the examples set into k subsets of approximately equal size. Then the learning process is run k times, using as training examples set k-1 of the subsets, and testing with the remaining subset (this subset is different for each of the k runs). In this way, a high classification rate due to a lucky selection of the test data is prevented. Moreover, as evolutionary algorithms are non-deterministic, one run is not meaningful. Thus, for each partition of the five-fold cross-validation, 10 runs were executed.

In each table, the first column indicates the used method. The coding is the following:

- The selection method: tournament (T) or uniform (U) selection (Section 5.2).
- The aggregation function (Eq. (26)): maximum (M) or addition (A).
- The rule weights (δ_i, Eq. (25)): no weights (N) or $\delta_i = confidence_i$ (W).
- The execution of the rule subset selection (Section 5.7) with the ILS algorithm (+S).

The other columns represent the number of rules (#*R*), the number of false positives (#*fp*) and negatives (#*fn*), and the percentage of examples that have been correctly classified (%correct), for both training and test sets. For each of these columns, three values are represented: $\overline{\chi}$, σ_{cv}, and $\overline{\sigma}_{run}$ [50]. $\overline{\chi}$ is the arithmetic mean over 50 executions (five-fold cross-validation with 10 runs). σ_{cv} is the standard deviation over the arithmetic means of each data partition, and represents the robustness of the

Fig. 9. Environments of the two datasets. (a) Ten people moving in groups in environment *uFrei*. (b) Several people standing still or moving at environment *uscCoie*.

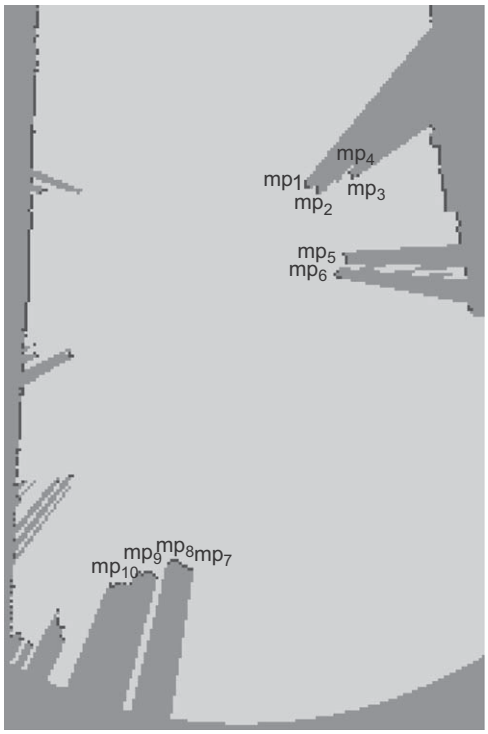

Fig. 10. Pattern modification.

algorithm to obtain similar results regardless the data partition. Finally, $\overline{\sigma}_{run}$ is the arithmetic mean of the standard deviations over the 10 runs for all the data partitions, which reflects the robustness of the probabilistic algorithm to obtain similar results regardless the followed pseudo-random sequence.

The values that have been used for the parameters of the evolutionary algorithm are: $maxIterations = 100$, $pop_{maxSize} = 200$, $p_c = 0.9$, $\alpha = 1$ (PCBLX crossover), $\upsilon = 5$ (Eq. (20)), $maxRestarts = 2$ (Fig. 8), and $radius_{nbhood} = 1$ (ILS algorithm). The best test result (%correct) with and without the rule subset selection for each dataset has been marked in bold in Tables 3 and 4. %correct has been calculated with Eq. (27).

For dataset *uscCoie* (Table 3), all the methods without rule selection obtain a good performance, and the best result on the test set corresponds to *TAW*. As expected, the methods using tournament selection overcome their counterparts with uniform selection, and the same occurs with the aggregation function (addition over maximum) and the use of weights. Rule subset selection slightly increments %correct in most of the methods and, again, *TAW*+*S* is the best one on the test set. However, the two best methods without rule selection obtain slightly worse results when rule selection is implemented. Of course, %correct increases on the training set but, due to overfitting, the performance on the test set is below the performance of the complete rule set. The algorithm shows a high robustness with the data partition (low value of σ_{cv}) and the different runs (low value of $\overline{\sigma}_{run}$).

Table 4 shows the results for dataset *uFrei*. As there is only one class (*mp*), the aggregation function does not influence the results and, therefore, only one aggregation function (addition) was analyzed. The inclusion of weights modifies the results, as if $\delta_i \approx 0$, then the association degree (Eq. (25)) is very small and the rule does not contribute to the classification. For the methods without selection, those using weights (W) are better, as the rules with a low δ_i, which means low confidence, are discarded. Again, the best method is *TAW*. When rule selection is included, %correct increases both for training and test. After rule selection, the performance of all the methods becomes very similar, as the selection algorithm eliminates the bad rules. *TAW*+*S* and *TAN*+*S* obtain the best result, as they exactly keep the same rules. $\overline{\sigma}_{run}$ has again a low value, however $\overline{\sigma}_{cv}$ is high. The reason is that most of the examples involving movement of the robot are contained in two of the data partitions and, moreover, in one of those partitions there is a high concentration of people, very close and in several groups that merge or split. Therefore, %correct decreases when these partitions are not included in the training set. This is a consequence of the difficulties for the detection when the robot is moving with a high concentration of people, and highlights the importance of this kind of situations in the training set in order to get robust and general classifiers.

Results show that the performance of the classifiers over the test sets in nearly all the methods is similarly good. Only in one case the performance was around a 10% lower than the average. This reflects that the system is reliable and robust independently of the combination of choices for the evolutionary algorithm and the fuzzy reasoning mechanism. Moreover, the values of other typical parameters of evolutionary algorithms, as crossover or mutation probabilities, have been selected using standard common values that work well in most cases instead of looking for very specific values for the proposed approach.

In comparison with the performance criteria used in [5,6], our criterion is much more demanding, as we do not consider negative examples for n_{ex}, i.e., a segment that is not a person does not contribute to n_{ex}, but if it is classified as a person it increments #*fp* and, consequently, reduces %correct. Using our

Table 3
Results of the five-fold cross-validation for dataset *uscCoie*.

Method	#R			Training									Test								
				#fp			#fn			%correct			#fp			#fn			%correct		
	$\overline{\chi}$	σ_{cv}	$\overline{\sigma}_{run}$	$\overline{\chi}$	σ_{cv}	$\overline{\sigma}_{run}$	$\overline{\chi}$	σ_{cv}	$\overline{\sigma}_{run}$	$\overline{\chi}$	σ_{cv}	$\overline{\sigma}_{run}$	$\overline{\chi}$	σ_{cv}	$\overline{\sigma}_{run}$	$\overline{\chi}$	σ_{cv}	$\overline{\sigma}_{run}$	$\overline{\chi}$	σ_{cv}	$\overline{\sigma}_{run}$
TMN	83.7	7.7	5.7	34.5	12.5	9.9	1.2	0.4	1.6	95.99	1.43	1.17	25.8	7.0	4.1	13.3	4.7	3.7	82.41	4.71	2.42
TMW	84.0	7.7	6.8	10.9	4.2	4.3	2.2	0.8	1.2	98.52	0.54	0.50	19.5	4.8	3.9	13.8	4.6	3.7	85.04	4.19	2.25
TAN	83.6	7.6	5.8	33.6	12.4	9.2	1.2	0.3	1.5	96.09	1.42	1.09	25.1	6.8	4.1	13.3	4.4	3.8	82.69	4.44	2.56
TAW	84.0	7.7	6.8	10.9	4.2	4.3	2.2	0.8	1.2	98.52	0.54	0.50	19.2	4.7	3.9	13.7	4.6	3.7	**85.20**	**4.14**	**2.26**
UMN	82.7	7.7	6.2	24.7	11.5	8.3	1.0	0.8	1.5	97.11	1.36	0.92	25.2	6.7	4.0	14.9	4.7	4.1	81.94	4.82	2.64
UMW	83.1	7.5	5.8	7.3	4.3	5.7	2.1	1.1	1.5	98.95	0.60	0.63	19.5	4.3	4.1	15.1	4.7	4.2	84.43	4.11	2.81
UAN	82.6	7.7	6.2	24.9	12.0	9.4	1.0	0.8	1.5	97.09	1.43	1.04	24.8	6.5	3.9	14.6	5.0	3.7	82.27	4.86	2.35
UAW	83.1	7.5	5.8	7.3	4.3	5.7	2.1	1.1	1.5	98.95	0.60	0.63	19.1	4.5	3.9	14.9	4.9	3.8	84.72	4.27	2.54
TMN + S	77.3	6.1	6.1	3.6	1.3	1.7	6.4	1.9	2.4	98.88	0.34	0.23	19.3	4.7	4.2	15.0	5.1	4.1	84.57	4.34	2.50
TMW + S	78.5	6.1	7.7	3.8	1.3	1.7	5.3	1.5	1.8	98.99	0.31	0.21	18.8	4.6	3.8	14.7	4.8	4.0	84.92	4.18	2.36
TAN + S	77.2	5.7	6.7	3.8	1.5	2.0	6.1	1.5	2.6	98.89	0.34	0.25	19.0	4.6	4.2	14.9	4.9	4.2	84.75	4.15	2.64
TAW + S	78.5	6.1	7.7	3.8	1.3	1.7	5.3	1.5	1.8	98.99	0.31	0.21	18.5	4.5	3.9	14.6	4.8	4.0	**85.07**	**4.14**	**2.40**
UMN + S	78.9	6.3	7.0	2.8	1.1	1.7	4.1	1.9	2.0	99.23	0.30	0.24	19.2	4.9	3.8	15.6	4.8	4.1	84.32	4.48	2.65
UMW + S	80.3	6.6	6.6	3.2	1.3	1.7	3.6	1.6	1.9	99.24	0.30	0.24	19.0	4.7	4.2	15.4	4.9	4.3	84.52	4.38	2.87
UAN + S	82.6	7.7	6.2	24.9	12.0	9.4	1.0	0.8	1.5	97.09	1.43	1.04	24.8	6.5	3.9	14.6	5.0	3.7	82.27	4.86	2.35
UAW + S	80.3	6.6	6.6	3.2	1.3	1.7	3.6	1.6	1.9	99.24	0.30	0.24	18.6	4.9	3.9	15.2	5.1	4.0	84.79	4.56	2.61

Table 4
Results of the five-fold cross-validation for dataset *uFrei*.

Method	#R			Training									Test								
				#fp			#fn			%correct			#fp			#fn			%correct		
	$\overline{\chi}$	σ_{cv}	$\overline{\sigma}_{run}$	$\overline{\chi}$	σ_{cv}	$\overline{\sigma}_{run}$	$\overline{\chi}$	σ_{cv}	$\overline{\sigma}_{run}$	$\overline{\chi}$	σ_{cv}	$\overline{\sigma}_{run}$	$\overline{\chi}$	σ_{cv}	$\overline{\sigma}_{run}$	$\overline{\chi}$	σ_{cv}	$\overline{\sigma}_{run}$	$\overline{\chi}$	σ_{cv}	$\overline{\sigma}_{run}$
TAN	88.4	11.5	6.9	137.8	29.3	23.5	0.5	0.4	0.6	87.18	2.66	2.18	67.4	71.3	8.5	9.0	4.6	2.2	71.82	25.04	3.13
TAW	88.4	11.5	6.9	28.9	3.6	5.0	17.9	4.5	3.3	95.66	0.64	0.45	39.1	46.7	6.1	13.0	6.5	2.1	**80.72**	**15.70**	**2.61**
UAN	89.4	12.2	5.4	91.5	16.2	19.7	0.3	0.2	0.4	91.50	1.49	1.84	57.4	62.8	6.2	8.2	4.4	2.2	75.75	21.93	2.54
UAW	89.4	12.2	5.4	26.5	2.7	4.7	12.9	4.0	3.2	96.35	0.57	0.43	41.7	49.5	5.2	11.7	5.7	1.9	80.25	16.74	2.09
TAN + S	69.4	11.3	8.1	7.7	2.4	3.3	28.8	5.2	3.9	96.62	0.49	0.30	33.2	39.1	6.6	14.9	6.0	1.8	**82.22**	**12.95**	**2.71**
TAW + S	69.4	11.3	8.1	7.7	2.4	3.3	28.8	5.2	3.9	96.62	0.49	0.30	33.2	39.1	6.6	14.9	6.0	1.8	**82.22**	**12.95**	**2.71**
UAN + S	73.1	10.5	6.9	7.1	1.8	4.1	23.5	4.0	4.5	97.16	0.44	0.29	35.2	43.7	5.1	13.6	6.3	1.8	81.94	14.58	2.17
UAW + S	73.1	10.5	6.9	7.0	1.6	4.0	23.6	4.1	4.4	97.16	0.44	0.29	35.1	43.8	5.1	13.6	6.3	1.8	81.95	14.58	2.18

performance measure (Eq. (27)), the results presented in [5] over two datasets get a value of %correct = 66% which is inferior to our performance. Moreover the datasets in [5] did not contain data with the robot moving or in populated environments (but in cluttered environments). In [6] the performances are provided as percentages, but the number of true positives, false positives, etc. are not indicated. Thus, a numerical comparison is not possible. However, authors indicate that their results with the leg detector are worse than those presented in [5], probably due to the complexity of their office dataset. In summary, our proposal obtains a much better performance than other existing approaches in spite of the complexity of the used datasets: people moving in groups combined with the movement of the robot, and also the distinction between static and moving people.

7. Conclusions

We described in this paper a pattern classifier system for the detection of people using laser range finders data. The two real environments in which the system was tested have several characteristics that make them specially difficult: people moving in groups, the robot was moving in part of the experiments, and the classifier had to distinguish between static and moving people. The spatio-temporal pattern analysis of the datasets of this problem was accomplished using the quantified fuzzy temporal rules (QFTRs) model. Moreover, the classifiers were learned through a genetic algorithm, and a deep experimental setup was done, including five-fold cross-validation and 10 runs per data partition. Some statistical measures of the results were calculated, reflecting a very high classification rate in quite complex and realistic conditions. These results reflect the ability of QFTRs for spatio-temporal pattern recognition, and that their combination with evolutionary algorithms is a powerful tool for pattern classification.

References

[1] J. Han, B. Bhanu, Fusion of color and infrared video for moving human detection, Pattern Recognition 40 (2007) 1771–1784.
[2] K. Huang, L. Wang, T. Tan, S. Maybank, A real-time object detecting and tracking system for outdoor night surveillance, Pattern Recognition 41 (2008) 432–444.
[3] T. Lam, R. Lee, D. Zhang, Human gait recognition by the fusion of motion and static spatio-temporal templates, Pattern Recognition 40 (2007) 2563–2573.
[4] S. Lu, J. Zhang, D. Feng, Detecting unattended packages through human activity recognition and object association, Pattern Recognition 40 (2007) 2173–2184.
[5] K. Arras, O. Martínez Mozos, W. Burgard, Using boosted features for the detection of people in 2D range data, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Rome, Italy, 2007, pp. 3402–3407.

[6] Z. Zivkovic, B. Kröse, Part based people detection using 2D range data and images, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego, USA, 2007, pp. 214–219.
[7] D. Schulz, W. Burgard, D. Fox, A. Cremers, People tracking with mobile robots using sample-based joint probabilistic data association filters, International Journal of Robotics Research 22 (2) (2003) 99–116.
[8] D. Amarasinghe, G. Mann, R. Gosine, Moving object detection in indoor environments using laser range data, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Beijing, China, 2006, pp. 802–807.
[9] M. Bennewitz, W. Burgard, G. Cielniak, S. Thrun, Learning motion patterns of people for compliant robot motion, International Journal of Robotics Research 24 (1) (2005) 31–48.
[10] J. Bobruk, D. Austin, Laser motion detection and hypothesis tracking from a mobile platform, in: Proceedings of Australasian Conference on Robotics and Automation (ACRA), Canberra, Australia, 2004.
[11] P. Chakravarty, R. Jarvis, Panoramic vision and laser range finder fusion for multiple person tracking, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Beijing, China, 2006, pp. 2949–2954.
[12] J. Cui, H. Zha, H. Zhao, R. Shibasaki, Robust tracking of multiple people in crowds using laser range scanners, in: Proceedings of the 18th International Conference on Pattern Recognition (ICPR), Hong Kong, China, 2006, pp. 857–860.
[13] M. Lindström, J.-O. Eklundh, Detecting and tracking moving objects from a mobile platform using a laser range scanner, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Maui, USA, 2001, pp. 1364–1369.
[14] E. Topp, H. Christensen, Tracking for following and passing persons, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Edmonton, Canada, 2005, pp. 2321–2327.
[15] C.-C. Wang, C. Thorpe, Simultaneous localization and mapping with detection and tracking of moving objects, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Washington, DC, USA, 2002, pp. 2918–2924.
[16] H. Zhao, Y. Chen, X. Shao, K. Katabira, R. Shibasaki, Monitoring a populated environment using single-row laser range scanners from a mobile platform, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Roma, Italy, 2007, pp. 4739–4745.
[17] J. Xavier, M. Pacheco, D. Castro, A. Ruano, U. Nunes, Fast line, arc/circle and leg detection from laser scan data in a player driver, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Barcelona, Spain, 2005, pp. 3930–3935.
[18] N. Bellotto, H. Hu, Multisensor integration for human-robot interaction, Intelligent Cybernetic Systems Journal 1(2005).
[19] B. Kluge, C. Köhler, E. Prassler, Fast and robust tracking of multiple moving objects with a laser range finder, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Seoul, Korea, 2001, pp. 1683–1688.
[20] A. Brooks, S. Williams, Tracking people with networks of heterogeneous sensors, in: Proceedings of the Australian Conference on Robotics and Automation (ACRA), Brisbane, Australia, 2003.
[21] C. Martin, E. Schaffernicht, A. Scheidig, H.-M. Gross, Sensor fusion using a probabilistic aggregation scheme for people detection and tracking, in: Proceedings of the 2nd European Conference on Mobile Robots (ECMR), Ancona, Italy, 2005, pp. 176–181.
[22] G. Taylor, L. Kleeman, A multiple hypothesis walking person tracker with switched dynamic model, in: Proceedings of the Australasian Conference on Robotics and Automation (ACRA), Canberra, Australia, 2004.
[23] P. Cariñena, A. Bugarín, M. Mucientes, F. Díaz-Hermida, S. Barro, Fuzzy temporal rules: a rule-based approach for fuzzy temporal knowledge representation and reasoning, in: Technologies for Constructing Intelligent Systems 2: Tools, Studies in Fuzziness and Soft Computing, vol. 90, Physica-Verlag, Wurzburg, 2002, pp. 237–50.
[24] P. Cariñena, A. Bugarín, M. Mucientes, S. Barro, A language for expressing fuzzy temporal rules, Mathware and Soft Computing 7 (2–3) (2000) 213–227.
[25] F. Herrera, Genetic fuzzy systems: taxonomy, current research trends and prospects, Evolutionary Intelligence 1 (2008) 27–46.
[26] H. Lourenço, O. Martin, T. Stützle, Iterated local search, in: Handbook of Metaheuristics, Kluwer Academic Publishers, Dordrecht, 2003, pp. 321–353.
[27] M. Mucientes, W. Burgard, Multiple hypothesis tracking of clusters of people, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Beijing, China, 2006, pp. 692–697.
[28] D. Glas, T. Miyashita, H. Ishiguro, N. Hagita, Laser tracking of human body motion using adaptive shape modeling, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego, USA, 2007, pp. 602–608.
[29] M. Kleinehagenbrock, S. Lang, J. Fritsch, F. Lömker, G. Fink, G. Sagerer, Person tracking with a mobile robot based on multi-modal anchoring, in: Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication (ROMAN), Berlin, Germany, 2002, pp. 423–429.
[30] M. Scheutz, J. McRaven, G. Cserey, Fast, reliable, adaptive, bimodal people tracking for indoor environments, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 2004, pp. 1347–1352.
[31] C. Martin, E. Schaffernicht, A. Scheidig, H.-M. Gross, Multi-modal sensor fusion using a probabilistic aggregation scheme for people detection and tracking, Robotics and Autonomous Systems 54 (2006) 721–728.
[32] C.-C. Wang, C. Thorpe, A. Suppe, Ladar-based detection and tracking of moving objects from a ground vehicle at high speeds, in: Proceedings of the Intelligent Vehicles Symposium, Columbus, USA, 2003, pp. 416–421.
[33] E. Topp, D. Kragic, P. Jensfelt, H. Christensen, An interactive interface for service robots, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), New Orleans, USA, 2004, pp. 3469–3474.
[34] H. Moravec, Sensor fusion in certainty grid for mobile robots, AI Magazine 9 (2) (1988) 61–74.
[35] F. Lu, E. Milios, Robot pose estimation in unknown environments by matching 2D range scans, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1994, pp. 935–938.
[36] M. Mucientes, R. Iglesias, C. Regueiro, A. Bugarín, P. Cariñena, S. Barro, Fuzzy temporal rules for mobile robot guidance in dynamic environments, IEEE Transactions on Systems, Man and Cybernetics—Part C: Applications and Reviews 31 (3) (2001) 391–398.
[37] M. Mucientes, R. Iglesias, C. Regueiro, A. Bugarín, S. Barro, A fuzzy temporal rule-based velocity controller for mobile robotics, Fuzzy Sets and Systems 134 (2003) 83–99.
[38] O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena, Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases, Advances in Fuzzy Systems-Applications and Theory, vol. 19, World Scientific, Singapore, 2001.
[39] B. Carse, T. Fogarty, A. Munro, Evolving temporal fuzzy rule-bases for distributed routing control in telecommunication networks, in: Genetic Algorithms and Soft Computing, Studies in Fuzziness and Soft Computing, vol. 8, Physica-Verlag, Wurzburg, 1996, pp. 467–488.
[40] T. Kovacs, Strength or Accuracy: Credit Assignment in Learning Classifier Systems, Springer, Berlin, 2004.
[41] O. Cordón, F. Herrera, Hybridizing genetic algorithms with sharing scheme and evolution strategies for designing approximate fuzzy rule-based systems, Fuzzy Sets and Systems 118 (2001) 235–255.
[42] A. Giornada, F. Neri, Search-intensive concept induction, Evolutionary Computation 3 (4) (1995) 375–416.
[43] D. Greene, S. Smith, Competition-based induction of decision models from examples, Machine Learning 3 (1993) 229–257.
[44] M. Wong, W. Lam, K. Leung, P. Ngan, J. Cheng, Discovering knowledge from medical databases using evolutionary algorithms, IEEE Engineering in Medicine and Biology Magazine 19 (4) (2000) 45–55.
[45] K. Leung, Y. Leung, L. So, K. Yam, Rule learning in expert systems using genetic algorithm: 1, concepts, in: Proceedings of the 2nd International Conference on Fuzzy Logic and Neural Networks, Iizuka, Japan, 1992, pp. 201–204.
[46] F. Herrera, M. Lozano, A. Sánchez, A taxonomy for the crossover operator for real-coded genetic algorithms: an experimental study, International Journal of Intelligent Systems 18 (2003) 309–338.
[47] M. Lozano, F. Herrera, N. Krasnogor, D. Molina, Real-coded memetic algorithms with crossover hill-climbing, Evolutionary Computation 12 (2004) 273–302.
[48] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer, Berlin, 1996.
[49] O. Cordón, M. del Jesus, F. Herrera, A proposal on reasoning methods in fuzzy rule-based classification systems, International Journal of Approximate Reasoning 10 (1999) 21–45.
[50] J. Casillas, O. Cordón, M. del Jesus, F. Herrera, Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction, IEEE Transactions on Fuzzy Systems 13 (1) (2005) 13–29.

About the Author—MANUEL MUCIENTES received the M.Sc. and Ph.D. degrees in Physics from the University of Santiago de Compostela, Spain, in 1998 and 2002, respectively. He is currently a *Ramón y Cajal* research fellow with the Department of Electronics and Computer Science, University of Santiago de Compostela. His current research interests include evolutionary algorithms, genetic fuzzy systems (specially for mobile robotics), visual SLAM (Simultaneous Localization and Mapping), spatio-temporal pattern recognition, and multitarget-multisensor tracking.

About the Author—ALBERTO J. BUGARÍN DIZ received the B.S. and Ph.D. degrees in physics from the University of Santiago de Compostela, Spain in 1990 and 1994, respectively. He is currently an Associate Professor at the Department of Electronics and Computer Science at the same university. His research interests focus on fuzzy temporal knowledge representation and reasoning, automated knowledge discovery and their applications in mobile robotics and industrial processes.