International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems
© World Scientific Publishing Company

FUZZY DISTANCE SENSOR DATA INTEGRATION AND INTERPRETATION

ZOE FALOMIR, VICENT CASTELLÓ, M. TERESA ESCRIG, JUAN CARLOS PERIS
Cognitive Robots (C-Robots), Espaitec, University Jaume I,
Castellón, E-12071, Spain
{*zfalomir, vcastello, mtescrig, jcperis* }*@c-robots.com*

Received (received date)
Revised (revised date)

An approach to distance sensor data integration that obtains a robust interpretation of the robot environment is presented in this paper. This approach consists in obtaining patterns of fuzzy distance zones from sensor readings; comparing these patterns in order to detect non-working sensors; and integrating the patterns obtained by each kind of sensor in order to obtain a final pattern that detects obstacles of any sort. A dissimilarity measure between fuzzy sets has been defined and applied to this approach. Moreover, an algorithm to classify orientation reference systems (built by corners detected in the robot world) as *open* or *closed* is also presented. The final pattern of fuzzy distances, resulting from the integration process, is used to extract the important reference systems when a glass wall is included in the robot environment. Finally, our approach has been tested in an ActivMedia Pioneer 2 dx mobile robot using the Player/Stage as the control interface and promising results have been obtained.

Keywords: Sensor data integration; fuzzy set theory; qualitative reasoning; robotics.

1. Introduction

Human beings use many kinds of sensory information (sight, hearing, smell, taste, touch, etc.) in order to obtain a complete and reliable representation of their surroundings. Results of neuroscience studies [1] explain that the information captured by human senses is perceived by the cerebral cortex as spatiotemporal patterns that are stored as memories. In the same way, robots can incorporate different kinds of sensors -each one sensitive to a different property of the environment- whose data could be integrated to make the perception of the robot more robust and to obtain new information, otherwise unavailable. Information captured through the robot sensors can be expressed as patterns of concepts, by transforming numerical data obtained into qualitative terms, and then, these patterns can be stored in a knowledge base.

Some robot sensors, such as laser and sonar sensors, capture the same physical magnitude of the environment that is the distance to objects. Although these kinds of sensors are sensitive to different properties of the environment (light and sound properties, respectively), they present different drawbacks which can be overcome by integrating or fusing data obtained by them. If no method to combine data provided by all the robot sensors is effectively used, important information about the robot environment may be lost. This is the reason why sensor data integration is important for robot navigation, as it provides the robot with knowledge about its surroundings that can help it to carry out a task successfully and more efficiently.

Finally, observations in robot-oriented industrial applications [2] emphasize that robots should have the capabilities of integrating reasoning, perception and action with conventional industrial tasks. In order to achieve such capabilities, the qualitative representation in a robotic system is required to have a natural connection to its quantitative representation and it also provides the atomic representation that could be used to build higher level cognitive functions for robots to enable them to reason, act, and perceive in dynamic, partially unknown, and unpredictable environments. Our research work is a little step to contribute in this direction.

2. Related Work on Distance Integration

Sensor data fusion represents the process of combining data or information from multiple sensors for detecting obstacles, object recognition, tracking of objects, etc. Generally, in the literature there are many approaches that deal with the problem of distance sensor data integration by using different probabilistic methods and incorporating different kinds of sensors. These methods obtain a high precision in their application but at a high computational cost, and they usually obtain a description of the world that has a higher degree of precision than that required by the task to be performed by the robot.

Sonar and laser data integration for mobile robot navigation has been characterized by the use of probabilistic techniques: covariance intersections [3], grid maps and Bayes' theorem [4], grid maps and the Dezert-Smarandache theory [5], Kalman filters [6], [7], Gauss approximation methods and vector maps [8], fuzzy segment maps [9], etc.

However, the extraction of knowledge from the world by numerical methods is very limited. A later interpretation of the coordinates where the robot is located is needed so that the robot can extract knowledge from the numerical values obtained. If the aim of sensor data integration is not localizing the robot accurately in the world, but extracting knowledge from it, qualitative representations are usually used. These representations define qualitative concepts for each important characteristic to distinguish in the world that can be used later on in decision processes.

In the literature, qualitative representations of sensor data have been applied to very few sensor data fusion approaches. We have found only Reece and Durrant-Whyte's works [10], [11], [12], [13] which mainly obtain qualitative descriptions of sensor cues. In their earlier work [10], they extracted regions of constant depth (RCD) from sonar sensor cues and interpret qualitatively the evolution of these RCD as the robot moves through the environment in order to identify planes, edges and corners. This work was extended [11], [12] in order to classify the surface curvature of the robot environment as convex, concave, plane, close-concave, far-concave, etc. and by combining a qualitative model based on intervals and qualitative differential equations (QDE) with a Kalman filter in order to interrelate the values of sensor observations with the system parameters and to estimate the parameters of the system for noisy processes or when the models obtained are incomplete or imprecise. Finally, another Reece's work[13] used qualitative descriptions of sensor cues for image understanding. A sensor cue contains a qualitative descriptor of the tool that processed the image, a qualitative representation of the spectral bands of the observed image -green, red, near-infrared or none of these- and a label denoting the interpretation of the representation. A qualitative reasoning system was built in order to distinguish soil from water in thermal daytime and nighttime images.

The motivation of our approach for distance sensor data integration is not localizing the robot accurately in the world, not interpreting sensor cues. Our main aim is to extract information about distances in the robot world by means of qualitative concepts that improve the knowledge of the robot of its surroundings and which could be used in later decision processes. Moreover, this qualitative information about distances is also used to detect obstacles in a robust way and also to characterize the obstacles or landmarks found in the world. Moreover, the semantic meaning of these qualitative names could be improved and related to others in the future by means of an ontology, as it has been previously done by the authors in their approach for generating ontology-based qualitative description of images[17].

A more recent research trend in literature is integrating results from distance sensor fusion with images taken from a camera in order to extract knowledge from the environment by means of an XML dataset [14], an ontology [15], symbolic/qualitative information [16], etc. This is the direction of our approach. However, first our aim is to extract knowledge from distance data fusion in an earlier stage and, later on, integrating it with the knowledge extracted from images by our approach based on description logics[17].

Our approach for distance sensor data integration (1) obtains patterns of fuzzy distances for each kind of distance sensor incorporated in a robot; (2) compares the patterns obtained in order to detect sensor errors; (3) integrates the patterns coming from different kinds of sensors to overcome the drawbacks presented individually and to obtain a more reliable perception of the environment; (4) provides knowledge to the robot by means of qualitative terms that categorize the distance of the robot to the obstacles and also types of obstacles; (5) can be extended and generalized for any kind of distance sensor and any kind of robot that includes distance sensors.

To the best of our knowledge, there is no approach for distance sensor data integration that presents these characteristics so it is not possible to carry out any comparative study with our approach.

3. Integration of Distance Sensor Data and Interpretation

Our approach for *Distance Sensor Data Integration and Interpretation* consists of the following steps, connected as Figure 1 shows.

(i) **Obtaining Robust Fuzzy Distance Patterns (FDPs)** from each kind of distance sensor (described in Section 4), which involves:
   (a) Transforming the distance sensor readings into patterns of fuzzy distance zones.
   (b) Comparing the patterns of fuzzy distance zones obtained in order to detect those sensors that are not working properly.
   (c) Obtaining a robust fuzzy distance pattern for each kind of distance sensor.

(ii) Integrating patterns provided by each kind of sensor in order to overcome each sensor's disadvantages and to obtain a final distance pattern that can **detect any sort of obstacles** (explained in Section 5).

(iii) Calculating the discontinuities of distances in the final distance pattern and relating them to the corners detected by the approach by Peris and Escrig [18] in order to **classify Reference Systems (RSs) in the robot world as *open* or *closed*** (as explained in Section 6).

(iv) Defuzzifying the final distance pattern in order to provide the robot with a **smooth speed** depending on its frontal distance to the obstacle (described in Section 7).

Our approach can be extended and generalized for any kind of robot incorporating sonar and laser distance sensors and other kinds of distance sensors such as infrared an so on. In Section 8, the results of our tests on a real robot platform are detailed and, finally, in Section 9, our conclusions are explained.

4. Obtaining Robust Fuzzy Distance Patterns (*FDP*s)

One of the main objectives of our approach is obtaining a reliable *Fuzzy Distance Pattern* (FDP) for each kind of distance sensor on robot. In order to achieve this, first we obtain patterns of fuzzy distance zones from sensor numerical readings, as described in Section 4.1. Then we compare these patterns to detect sensors with technical problems, as described in Section 4.3. In order to compare fuzzy distance sets ($FdSets$), we have developed a $Dissimilarity\ Factor(DF)$, as explained in Section 4.2.

4.1. *Building FDPs*

In order to transform numerical distances obtained from the sensor readings into fuzzy distances, a fuzzy distance set is used. The concept of the fuzzy set was introduced by Zadeh [19] as *a 'class' with a continuum of grades of membership.* Since then they have become the foundation of a methodology for translating the

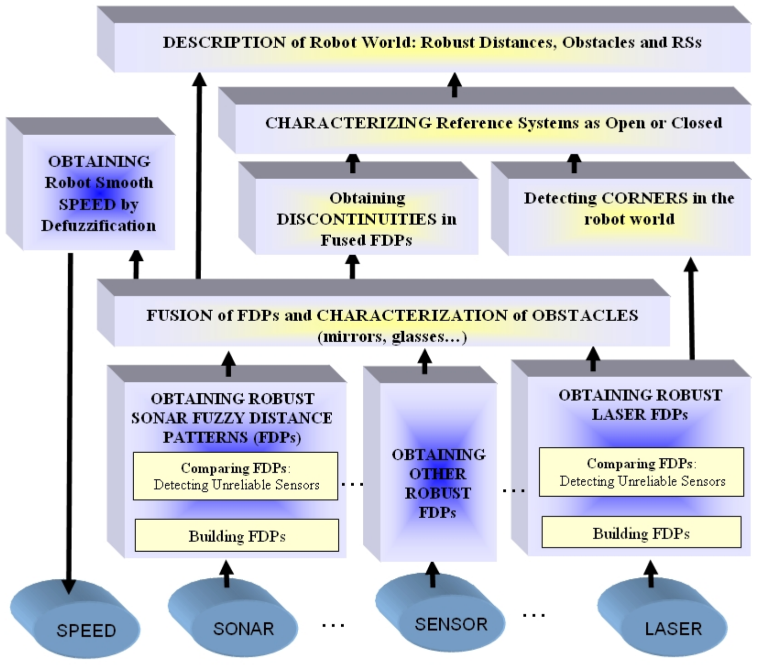

DESCRIPTION of Robot World: Robust Distances, Obstacles and RSs
CHARACTERIZING Reference Systems as Open or Closed
OBTAINING Robot Smooth SPEED by Defuzzification
Obtaining DISCONTINUITIES in Fused FDPs
Detecting CORNERS in the robot world
FUSION of FDPs and CHARACTERIZATION of OBSTACLES (mirrors, glasses...)
OBTAINING ROBUST SONAR FUZZY DISTANCE PATTERNS (FDPs)
Comparing FDPs: Detecting Unreliable Sensors
Building FDPs
OBTAINING OTHER ROBUST FDPs
OBTAINING ROBUST LASER FDPs
Comparing FDPs: Detecting Unreliable Sensors
Building FDPs
SPEED
SONAR
SENSOR
LASER

Fig. 1. Scheme of our approach for Distance Sensor Data Integration and Interpretation.

numerical data obtained from the world into linguistic categories or classes that can be given a meaning and used for reasoning.

In our approach, fuzzy distance values are used because (1) they provide linguistic values of distance, which can be given a meaning that can be useful to the robot later on for decision processes, and because (2) they can be easily defuzzified into the original numerical values.

Let us define a fuzzy set as a pair $(FdSet, \mu_{FdSet})$ where $FdSet$ is a set and $\mu_{FdSet} : FdSet \to [0, 1] \in \Re$. For a finite set $FdSet = \{x_1, ..., x_n\}$, for each $x \in FdSet$, a $\mu_{FdSet}(x)$ is obtained and called the grade of membership of $x \in (FdSet, \mu_{FdSet})$.

Let $x \in FdSet$. Then x is called not included in the fuzzy set $(FdSet, \mu_{FdSet})$ if $\mu_{FdSet}(x) = 0.0$, x is called fully included if $\mu_{FdSet}(x) = 1.0$, and x is called fuzzy member if $0.0 < \mu_{FdSet}(x) < 1.0$.

For each numerical distance obtained from the sensor readings, its grade of membership to each defined *Fuzzy distance Set* ($FdSet$) is calculated and those fuzzy distances obtained ($\{x_1, ..., x_n\}$) whose grade of membership is other than zero ($\mu_{FdSet}(x_i) \neq 0$) are selected. Those numerical sensor readings that are negative or exceed the sensor range are characterized as *out_of_range* distances with the maximum grade of membership (1.0).

After transforming all sensor readings into fuzzy distances, those qualitative distances with the same name are grouped into zones and fuzzy distance patterns are obtained.

Let us consider a *Fuzzy Distance Pattern* (FDP_t) as a collection of fuzzy distance zones related to the same sensor scan at a time *t*. Each *zone* includes its starting and ending angular position, the event corresponding to the zone and a list of fuzzy distances related to it. The grade of membership of these fuzzy distances is obtained as the mean of all the grades of membership originally included in the zone. And the rest of parameters are defined as:

$FDP_{time}(SensorType)$ =([Zone$_0$, ... ,Zone$_K$]).
$SensorType$ = {*sonar, laser, infrared, etc.*}
$Zone_i = [[Start, End], FdSet, Event]$
$FdSet = [(x_1, \mu_{FdSet}(x_1)), ..., (x_i, \mu_{FdSet}(x_i)), ..., (x_n, \mu_{FdSet}(x_n))]$
$Start \in [0, MaxAngularRange] \in N$
$End \in [0, MaxAngularRange] \in N$
$Event$ = {*simple_obstacle, glass_or_mirror, sound_reflection, SensorType_error*}

An example of a situation of a general robot with a common distance sensor inside a room and the corresponding FDP_t obtained is shown in Table 1. This $FDP_t(sensor)$ is made up of five *Zone*s ($K = 5$). The starting angular position of $Zone_1$ is $a_1°$ and its ending angular position is $a_2°$ and its angular amplitude is determined by $a_2 - a_1$. The fuzzy distances related to the first zone are determined by the distance names ($\{x_1, ..., x_n\}$) and grades of membership $\{\mu_{FdSet}(x_1), ..., \mu_{FdSet}(x_n)\}$ contained by $FdSet_{Zone_1}$. The sort of obstacle the robot is facing is determined by *Event*. The remaining zones of the pattern are described in the same way.

4.2. *Defining a Dissimilarity Factor (DF) between FdSets*

In order to compare *Fuzzy distance Sets* (*FdSet*s), a *Dissimilarity Factor* (*DF*) has been defined. This *DF* compares both the qualitative distances and their corresponding grades of membership.

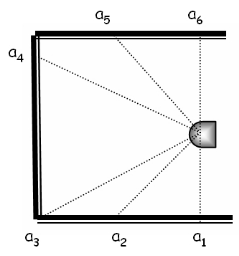
Given two general *FdSet*s, $FdSet_A$ and $FdSet_B$, containing n and m elements respectively:

$FdSet_A = \{A_1, A_2, ..., A_n\}$ where $A_i = (x_{A_i}, \mu_A(x_{A_i}))$
$FdSet_B = \{B_1, B_2, ..., B_m\}$ where $B_j = (x_{B_j}, \mu_B(x_{B_j}))$

The dissimilarity (*dSim*) between two elements, each corresponding to a different fuzzy set, A_i and B_j, is defined by:

$$dSim(A_i, B_j) = dSimQd(x_{A_i}, x_{B_j}) \cdot \mu_A(x_{A_i}) \cdot \mu_B(x_{B_j}) \qquad (1)$$

Table 1. An example of a situation of the robot in a room and the $FDP_t(sensor)$ obtained.

Robot Situation	FDP$_t$
	$FDP_t(sensor)$ = ([Zone$_1$, ... ,Zone$_5$]) =([[[a_1, a_2], [$FdSet_{Zone_1}$], Event], [[a_2+1, a_3], [$FdSet_{Zone_2}$], Event], [[a_3+1, a_4], [$FdSet_{Zone_3}$], Event], [[a_4+1, a_5], [$FdSet_{Zone_4}$], Event], [[a_5+1, a_6], [$FdSet_{Zone_5}$], Event]]).

Note that x_{A_i} corresponds to the fuzzy set name associated with fuzzy distance A_i, while $\mu_A(x_{A_i})$ is the grade of membership associated with the fuzzy set name x_{A_i}.

The dissimilarity between labels of qualitative distances (x_{A_i}, x_{B_j}) or *dSimQd* in Eq. 1 is solved by analyzing the conceptual neighborhood relations between the concepts defined. The term *conceptual neighborhood* was introduced by Freksa [21] in his analysis of the 13 interval relations defined in Allen's temporal logic [20]: *"Two relations between pairs of events are conceptual neighbors if they can be directly transformed one into another by continuous deformation (i.e., shortening or lengthening) of the events"*.

Conceptual neighborhood relations can be found between the qualitative labels defining distances. For example, the distances x_i (i.e. *far*) and x_{i+1} (i.e *very_far*) can be considered conceptual neighbors since a quantitative extension of the distance x_i leads to a direct transition to the distance x_{i+1}. However, the distances x_i and x_{i-2} (i.e. *close*) are not conceptual neighbors, since a transition between them must go through the distance x_{i-1} (i.e. *near*) first. Therefore, let us define the dissimilarity value between two qualitative distances that are conceptual neighbors as one positive unit if the first distance is smaller than the second distance and one negative unit otherwise:

$$x_1 \xrightarrow{+1} x_2 \xrightarrow{+1} x_3 \cdots x_i \xrightarrow{+1} x_{i+1} \cdots x_{n-2} \xrightarrow{+1} x_{n-1} \xrightarrow{+1} x_n$$
$$x_1 \xleftarrow[-1]{} x_2 \xleftarrow[-1]{} x_3 \cdots x_i \xleftarrow[-1]{} x_{i+1} \cdots x_{n-2} \xleftarrow[-1]{} x_{n-1} \xleftarrow[-1]{} x_n$$

All the possible dissimilarity values between general qualitative distances are calculated in Table 2.

Table 2. Dissimilarity matrix for qualitative distances.

	x_1	x_2	x_3	..	x_i
x_1	0	1	2	..	$i-1$
x_2	-1	0	1	..	$i-2$
x_3	-2	-1	0	..	$i-3$
..	..	..	..	..	..
x_j	$1-j$	$2-j$	$3-j$	..	$i-j$

The dissimilarity value between the same qualitative distances is zero ($dSimQd(x_{A_i}, x_{A_i}) = 0$) and the dissimilarity between the fuzzy distances defined as the limits of the set is the maximum dissimilarity, which is the cardinality of the defined $FdSet$ ($dSimQd(x_{A_i}, x_{A_n}) = card(FdSet_A) - 1$).

Finally, the number of dissimilarities among all the elements that is needed to calculate in order to obtain the final dissimilarity between two fuzzy sets, $FdSet_A$ and $FdSet_B$, are given by the Cartesian product of their elements ($n{\cdot}m$). Therefore, the $Dissimilarity\ Factor$ (DF) between two fuzzy sets is obtained by accumulating the dissimilarity value between each pair of elements that composes each relation obtained by the Cartesian product of the two sets involved, as Eq. 2 shows.

$$DF(FdSet_A, FdSet_B) = \sum dSim([A_1..A_n] \times [B_1..B_m]) \tag{2}$$

Considering $n = 2$ and $m = 3$,

$$FdSet_A = [A_1, A_2] = [[x_{A_1},\ \mu_A(x_{A_1})], [x_{A_2},\ \mu_A(x_{A_2})]]$$
$$FdSet_B = [B_1, B_2, B_3] = [[x_{B_1},\ \mu_B(x_{B1})], [x_{B2},\ \mu_B(x_{B_2})], [x_{B_3},\ \mu_B(x_{B_3})]]$$

Eq. 2 corresponds to:

$$\begin{aligned} DF(FdSet_A, FdSet_B) &= dSimQd(x_{A_1}, x_{B_1}) \cdot \mu_A(x_{A_1}) \cdot \mu_B(x_{B_1}) \\ &+\ dSimQd(x_{A_1}, x_{B_2}) \cdot \mu(x_{A_1}) \cdot \mu(x_{B_2}) \\ &+\ dSimQd(x_{A_1}, x_{B_3}) \cdot \mu(x_{A_1}) \cdot \mu(x_{B_3}) \\ &+\ dSimQd(x_{A_2}, x_{B_1}) \cdot \mu(x_{A_2}) \cdot \mu(x_{B_1}) \\ &+\ dSimQd(x_{A_2}, x_{B_2}) \cdot \mu(x_{A_2}) \cdot \mu(x_{B_2}) \\ &+\ dSimQd(x_{A_2}, x_{B_3}) \cdot \mu(x_{A_2}) \cdot \mu(x_{B_3}) \end{aligned}$$

4.3. *Comparing FDPs for detecting unreliable sensor readings*

In order to detect sensor malfunctions, the environment is scaned three times while the robot is static and the three fuzzy distance patterns obtained are compared

$(FDP_t, FDP_{t-1}, FDP_{t-2})$. Assuming that all the objects in the robot environment are also static, if a sensor obtains different readings depending on time and not on the situation, it is possible that this sensor has a technical problem.

The static comparison of patterns consists of comparing the current pattern (FDP_t) with the two previous ones (FDP_{t-1}, FDP_{t-2}) and determining which is the most new and reliable.

In order to compare fuzzy distance patterns, first a measure of similarity between the $Zones$ that build them must be defined. Therefore, let us consider that two $Zones$ are qualitatively similar $(Qsimilar)$ if the fuzzy distance sets ($FdSet$s) contained in them have a $DF = 0$ and that they are close in meaning $(CloseMeaning)$ if they have a $|DF| < 2$.

Therefore, two fuzzy distance patterns (FDP_t, FDP_{t-1}) are considered qualitatively similar $(QsimilarPatterns)$ if they are composed of the same number of $Zones$ which are qualitatively similar $(Qsimilar)$.

As Algorithm 1 shows, if the current pattern (FDP_t) and at least one of the two previous ones are $QsimilarPatterns$, the current pattern is selected as the final pattern $(FinalFDP_t)$. Otherwise, the two previous patterns are compared (FDP_{t-1}, FDP_{t-2}) and if they are $QsimilarPatterns$, the most recent pattern (FDP_{t-1}) is selected as the final one $(FinalFDP_t)$. However, if none of the patterns are completely $Qsimilar$, a new pattern is built.

Algorithm 1 Description of the static comparison of fuzzy distance patterns

```
From $Sensor(x)$ obtaining: $FDP_t$, $FDP_{t-1}$ and $FDP_{t-2}$
if $QsimilarPatterns(FDP_t, FDP_{t-1})$ or $QsimilarPatterns(FDP_t, FDP_{t-2})$
then
    $FinalFDP_t \leftarrow FDP_t$
else if $QsimilarPatterns(FDP_{t-1}, FDP_{t-2})$ then
    $FinalFDP_t \leftarrow FDP_{t-1}$
else
    $FinalFDP_t \leftarrow$ Building_new_$FDP(FDP_t, FDP_{t-1}, FDP_{t-2})$
end if
```

As Algorithm 2 shows, in order to build a new FDPs from the most reliable zones of the previous ones, for each angular position (a), the most current $FdSet$ that is $Qsimilar$ or have $CloseMeaning$ to the others is selected to build the $FinalFDP$.

If there is any angular position (a) where all the $FdSet$s are very different from each other, nothing about the real distance can be known. Therefore this reading is characterized as *none*, meaning no distance, with the maximum grade of membership. A value of *none* suggests technical problems with the sensor located in the angular position where the reading was taken.

Algorithm 2 Building a new FDP from the previous FDP obtained

```
for each Angular position($a$) in $FDP_t$, $FDP_{t-1}$ and $FDP_{t-2}$ do
  $FdSet_t \leftarrow Extract\_FdSet(FDP_t, a)$
  $FdSet_{t-1} \leftarrow Extract\_FdSet(FDP_{t-1}, a)$
  $FdSet_{t-2} \leftarrow Extract\_FdSet(FDP_{t-2}, a)$
  if $Qsimilar(FdSet_t, FdSet_{t-1})$ or $Qsimilar(FdSet_t, FdSet_{t-2})$ then
    $FdSet_{final} \leftarrow FdSet_t$
  else if $Qsimilar(FdSet_{t-1}, FdSet_{t-2})$ then
    $FdSet_{final} \leftarrow FdSet_{t-1}$
  else if $CloseMeaning(FdSet_t, FdSet_{t-1})$ or
  $CloseMeaning(FdSet_t, FdSet_{t-2})$ then
    $FdSet_{final} \leftarrow FdSet_t$
  else if $CloseMeaning(FdSet_{t-1}, FdSet_{t-2})$ then
    $FdSet_{final} \leftarrow FdSet_{t-1}$
  else
    $FdSet_{final} \leftarrow (none, 1.00)$
    $Event \leftarrow SensorType\_error$
  end if
  $FinalFDP_t \leftarrow Add\_FdSet(FdSet_{final}, a)$
end for
```

5. Integration of Sonar and Laser FDPs and Detection of Static Special Obstacles

As sonar and laser sensors have problems in different situations of the robot environment, their readings can be integrated to overcome these problems and also to identify the specific situation that the robot is facing.

The main problems of sonar sensors are: multiple reflections in corners; uncertainty in locating the target due to the cone-shaped beam; and external ultrasound sources or crosstalk. Although laser sensors usually provide accurate readings, they may also present some drawbacks related to the nature of the target surfaces. Low reflectance surfaces, like dark colors or soft materials, absorb the laser beam and return it with a feeble intensity; while high reflectance surfaces present more serious problems: mirrors reflect the laser beam in any direction, while glasses can react to the laser beam as transparent, partial mirrors or perfect mirrors, depending on the glass type, thickness and angle of incidence. Therefore, as these sensors fail in different situations, a method to extract the advantages of both of them can be developed.

In Algorithm 3, the integration of sonar and laser $FDPs$ is described. After obtaining a robust $FinalFDP$ from both sonar and laser sensors (Section 4.3), we check if any of the sensors has technical problems, that is, if they obtain distances defined as *none* or out_of_range. If both kinds of sensors show technical problems for the same $Zone$, the corresponding $Zone$ of the $FinalFDP$ will indicate

$Sonar\_Laser\_error$ as the $Event$. If, for the same $Zone$, one sensor has technical problems while the other one works well, the distance obtained by the second one is included in the $FinalFDP$ and the type of sensor that has technical problems ($sonar\_error$ or $laser\_error$) is indicated as the $Event$ for that $Zone$. If none of the sensors has technical problems in a $Zone$, then the sonar $FdSet$ and the laser $FdSet$ corresponding to that $Zone$ are compared by calculating a $Dissimilarity\ Factor$ (DF). If there is a large DF between both $FdSet$s, we can determine that:

- If the DF is larger than a threshold and positive, the distance obtained by the laser sensor is much larger than the sonar one. Therefore the robot could be facing a *glass window or mirror* that reflects the laser beam in any direction and this will be the $Event$ determined.
- If the DF is larger than a threshold and negative, the distance obtained by the sonar sensor is much larger than the laser one. Therefore the robot could be facing a corner that could have made the sound waves rebound and not return to the receiver. Then the $Event$ determined would be $sound\ reflection$ in corner.

If there is not a large DF between the $FdSet$s obtained by each type of sensor, then the final $FdSet$ for any angular position is that obtained by the laser sensor, since it is the most accurate sensor, and the $Event$ determined is *simple obstacle*.

Algorithm 3 Description of the integration of sonar and laser FDPs

```
for all $FdSet(sonar)$ in $FDP(Sonar)$ and $FdSet(laser)$ in $FDP(Laser)$ do
  if $FdSet(sonar)$ and $FdSet(laser)$ are $none$ then
    $FdSet(final), Event \leftarrow (none, 1.00)$, $Sonar\_Laser\_error$
  else if $FdSet(sonar)$ and $FdSet(laser)$ are $out\_of\_range$ then
    $FdSet(final), Event \leftarrow (out\_of\_range, 1.00), Sonar\_Laser\_out\_of\_range$
  else if $FdSet(laser)$ is $none$ or $out\_of\_range$ then
    $FdSet(final), Event \leftarrow FdSet(sonar), Laser\_error$
  else if $FdSet(sonar)$ is $none$ or $out\_of\_range$ then
    $FdSet(final), Event \leftarrow FdSet(laser), Sonar\_error$
  else
    $DF(FdSet(sonar), FdSet(laser))$
    if $|DF| \geq Threshold$ and $DF > 0$ then
      $FdSet(final), Event \leftarrow FdSet(sonar), glass\_or\_mirror$
    else if $|DF| \geq Threshold$ and $DF < 0$ then
      $FdSet(final), Event \leftarrow FdSet(laser), sound\_reflection$
    else
      $FdSet(final), Event \leftarrow FdSet(laser), simple\_obstacle$
    end if
  end if
  $FinalFDP \leftarrow FdSet(final), Event$
end for
```

6. Characterizing Reference Systems (RS) using the defined FDPs and DF

In this section, a DF is obtained between the pairs of $FdSet$s that compose the final FDP in order to discover if there are large discontinuities in the distances obtained. If these exist, it is supposed that there is an area that cannot be seen by the current position of the robot. This information combined with the corner information provided by Peris and Escrig's approach [18], enables the robot to characterize Reference Systems (RS) as *closed* or *open.*

6.1. *Overview of Peris and Escrig's approach for building RSs*

In Peris and Escrig's approach [18], the corners in a room that are detected by a robot are defined as the main landmarks of that room. Two consecutive corners (which can be concave or convex) in a robot scan define a Reference System (RS) and a set of RSs determines the final map of the room. Two kinds of RSs can compose a hybrid map defined in Peris and Escrig[18]: *closed* RS and *open* RS. *Closed* RSs are those in which a new landmark cannot appear between those landmarks, which are the limits of the RS and define it. However, *open* RSs are those in which new landmarks can appear between the two landmarks that define the RS and, as a consequence, a more accurate exploration is needed in order to define clearly all the main landmarks in the room.

As an example, let us consider the robot situation shown in Figure 2, where the robot has detected four corners: C1 and C4 are *concave*, while C2 and C3 are *convex*. By joining these consecutive corners, three RSs are obtained, RS_{12}, RS_{23} and RS_{34}. As shown in Figure 2 RS_{12} and RS_{34} are *open* while RS_{23} is *closed.* This characterization can also be inferred from the $FDPs$ obtained by our approach, as explained in Section 6.3.

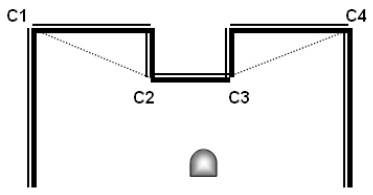


Fig. 2. Example of a robot situation inside a room

6.2. *Calculating Discontinuities in the Final FDP*

The final fuzzy distance pattern ($FinalFDP$) resulting from the integration of the sonar and laser $FDPs$ (see Section 5) can be used to classify RSs, defined by Peris and Escrig[18], as *open* or *closed*.

By calculating the $DissimilarityFactor$ (DF) between the $FdSets$ of each $Zone_i$ (simplified by Z_i) of the $FinalFDP$, large dissimilarities between $Zones$ can be detected, which correspond to discontinuities in the robot environment:

- If the DF is larger than a threshold ($|DF| >> T$) and negative ($DF < 0$), it corresponds to a change from a $Zone$ containing large fuzzy distances to a $Zone$ with small fuzzy distances, which is called an *approaching_discontinuity* in our approach.
- If the DF is larger than a threshold ($|DF| >> T$) and positive ($DF > 0$), it corresponds to a change from a $Zone$ containing small fuzzy distances to a $Zone$ with large fuzzy distances, which is called a *moving_away_discontinuity* in our approach.

An example of the possible discontinuities that our approach could obtain from the $FinalFDP$ in the general situation shown in Fig. 2 is the following:

$FDP(final)$=([	
[[a_1, a_2],[$FdSet_{Z_1}$]],	
[[a_2, a_3],[$FdSet_{Z_2}$]],	DF($FdSet_{Z_1}$, $FdSet_{Z_2}$)
...,	...
[[..., $a_{C}2$],[$FdSet_{Z_i}$]],	DF($FdSet_{Z_{i-1}}$, $FdSet_{Z_i}$)
[[$a_{C}2$, $a_{C}3$],[$FdSet_{Z_{i+1}}$]],	DF($FdSet_{Z_i}$, $FdSet_{Z_{i+1}}$) $\rightarrow |DF| >> T$ and $DF < 0$
[[$a_{C}3$, a_j],[$FdSet_{Z_{i+2}}$]],	DF($FdSet_{Z_{i+1}}$, $FdSet_{Z_{i+2}}$) $\rightarrow |DF| >> T$ and $DF > 0$
...,	...
[[a_k, a_{k+1}],[$FdSet_{Z_k}$]]).	DF($FdSet_{Z_{k-1}}$, $FdSet_{Z_k}$)

If we analyze the previous general distance pattern obtained by the robot when it is placed in the situation described in Figure 2, we observe that two large dissimilarities can be found:

- One between Zone$_i$ and Zone$_{i+1}$, reflected by a DF large and negative, which corresponds to an *approaching* discontinuity in the angular position where corner C_2 is located approximately ($a_{C}2$); and
- One between Zone$_{i+1}$ and Zone$_{i+2}$, reflected by a DF large and positive, which corresponds to a *moving away* discontinuity in the angular position where corner C_3 is located approximately ($a_{C}3$).

6.3. *Characterizing RSs as open or closed*

By relating the approximate angular location of the discontinuities obtained from the final FDP with the approximate angular location and the type of the corners (*concave* or *convex*) detected, an algorithm to classify the reference systems (RSs) of the robot environment as *open* or *closed* can be defined.

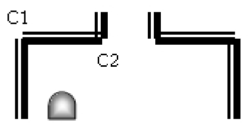
The main situations in which the robot can find *consecutive* corners are shown in Figure 3 and detailed next:

- **Situation 1**: a *concave* corner (C1) and a *convex* corner (C2),
  - (a) If a *moving away* discontinuity coincides with the *convex* corner (C2), if it happens after the *convex* corner, and then RS defined is *closed* (Situation 1a), but if it happens before the *convex* corner, and then RS defined is *open* (Situation 1b).
  - (b) If there are no discontinuities that coincide with these corners, they define a *closed* RS (Situation 1c).
- **Situation 2**: a *convex* corner (C1) and a *concave* corner (C2),
  - (a) If an *approaching* discontinuity coincides with the *convex* corner (C1), it happens after the *convex* corner, and then the RS defined is *closed* (Situation 2a), but if it happens before the *convex* corner, and then the RS defined is *open* (Situation 2b).
  - (b) If there are no discontinuities that coincide with these corners, they define a *closed* RS (Situation 2c). This situation is symmetrical to the previous one.
- **Situation 3**: two *convex* corners,
  - (a) If there is a *moving away* discontinuity coinciding with the first *convex* corner, the discontinuity happens after the corner, therefore the RS defined is *open* (Situation 3a).
  - (b) If there is an *approaching* discontinuity coinciding with the second *convex* corner, the discontinuity happens before the corner and the RS defined is *open* (Situation 3b).
  - (c) If an *approaching* discontinuity coincides with the first *convex* corner it happens before this corner and it corresponds to a previous RS. Similarly, if a *moving away* discontinuity coincides with the second *convex* corner, it happens after this corner and it corresponds to the following RS. This is the reason that although both convex corners coincide with discontinuities in situation 3c the RS determined by them is *closed.*
  - (d) If no discontinuity coincides with both convex corners, the RS defined by them is *closed.*
- **Situation 4**: Two *concave* corners. These corners always define a *closed* RS, since there cannot be a discontinuity of the robot environment between them, as a discontinuity involves the detection of another corner between the original ones.

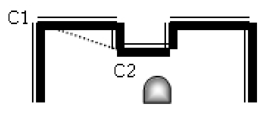
Finally, it is important to consider that the same discontinuity in the distance pattern cannot be related to more than one open reference system.

As Algorithm 4 shows, by analysing the previous situtations, it can be deduced that if the first corner is *convex* and coincides with a *moving away* discontinuity, this discontinuity takes place before the second corner and, therefore, they define

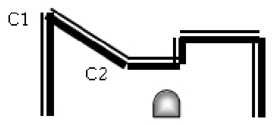
Fig. 3. Situations described related to the lines of Algorithm 4.



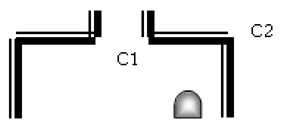
(a) Situation 1a



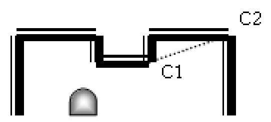
(b) Situation 1b



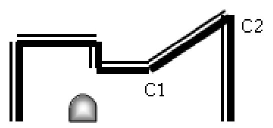
(c) Situation 1c



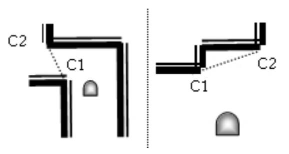
(d) Situation 2a



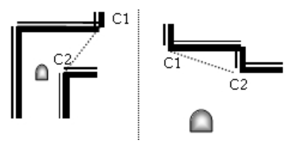
(e) Situation 2b



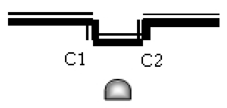
(f) Situation 2c



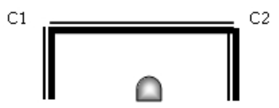
(g) Situation 3a



(h) Situation 3b



(i) Situation 3c



(j) Situation 4

an *open* RS. Similarly, if the second corner is a *convex* corner and coincides with an *approaching* discontinuity, this discontinuity also takes place before the second corner and, therfore, they define an *open* RS. In other situation, the RS determined by the two consecutive corners is *closed*.

Algorithm 4 Classification of *RS* as *open* or *closed*

```
if (C1.type = convex) and (discontinuity_in(C1) = moving_away) then
  SR(C1,C2).type ← open
else if (C2.type = convex) and (discontinuity_in(C2) = approaching) then
  SR(C1,C2).type ← open
else
  SR(C1,C2).type ← closed
end if
```

6.4. *Integrating Final FDP with RS information*

Because of its high precision, the corners in the robot world are obtained by analyzing the distances provided by the laser sensor.

However, when a glass surface is located in front of the robot, corners corresponding to the obstacles on the other side of the glass are detected. These corners are *false* corners that do not correspond to the real world. Therefore, in this situation, an integration of the information provided by the final FDP with the information of the RSs obtained is needed.

The integration done in our approach consists in not considering those corners whose angular distance coincides with the location of a *glass or mirror*, as Algorithm 5 shows.

Algorithm 5 Integration of the final FDP with the Corners information to build the real RS.

```
for $Corner(id, angle, type)$ in $CornersVector$[0 .. N] do
  if $angle$ not included in a $Zone_i$ with $Event \leftarrow Glass\_or\_mirror$ then
    $NewCornersVector \leftarrow Corner(id, angle, type)$
  end if
end for
$BuildNewRSs(NewCornersVector$[0 .. M]$)$
```

7. Defuzzification of FDPs to Obtain a Smooth Robot Speed

As our approach for distance sensor data integration obtains patterns of fuzzy distances, fuzzy set theory can be used in order to control the speed of the robot and to obtain smooth movements while the robot is approaching an obstacle.

Therefore, a fuzzy controller has been designed in order to define the robot speed depending on the frontal $Zone$ (90° aproximately) of the $FinalFDP$ resulting from the integration. This controller is composed of:

- the fuzzy distance set $FdSet$ representing distance to the obstacles: $(FdSet, \mu_{FdSet})$ where $FdSet$ is a finite set $FdSet = \{x_1, ..., x_n\}$ and $\mu_{FdSet} : FdSet \rightarrow [0,1] \in \Re$

- the fuzzy speed set $FSpeed$ representing the robot speed: $(FSpeed, \mu_{FSpeed})$ where $FSpeed$ is a finite set $FSpeed = \{y_1, ..., y_p\}$ and $\mu_{FSpeed} : FSpeed \rightarrow [0,1] \in \Re$

- a set of rules that relate each element of the fuzzy distance set $FdSet$ with the corresponding fuzzy speed set $FSpeed$: *if distance is* $x_i \in FdSet$ *then speed is* $y_r \in FSpeed$

- a defuzzification method which obtains the corresponding numerical speed from the obtained fuzzy distances. Our approach uses the Centre of Gravity (CoG) weighted by height for defuzzification, although other methods could also be used (more details are given by Galindo[22]).
  The rules defined relate the $FdSet$s obtained with the corresponding fuctions that define the $FSpeed$ sets. The selected functions of the $FSpeed$ set are truncated by height according to the grade of membership of the corresponding $FdSet$s and finally, the Centre of Gravity (CoG) of the area defined by those functions is obtained as the final numerical robot speed.
  The formula of the CoG is shown in Eq. 3, where $FSpeed_i(x)$ are the functions representing speed selected by the rules; and the formula of the CoG weighted by height is shown in Eq. 4, where H_i is the height to truncate $FSpeed$ functions and which correspond to the grade of membership obtained by the $FdSet$s in the frontal $Zone$ of the FDP (90° aproximately).

$$CoG_i = \frac{AreaX_i}{Area_i} = \frac{\int FSpeed_i(x) \cdot x \; dx}{\int FSpeed_i(x) \; dx} \tag{3}$$

$$\widehat{x} = \frac{\sum_{i=1}^{n} H_i}{\sum_{i=1}^{n} H_i \cdot CoG_i} \tag{4}$$

8. Experimentation

Our physical robotic platform was an ActiveMedia Pioneer 2 dx mobile robot[a] containing eight sonar sensors and a SICK LMS-200 laser range scanner [b]. As Figure 4 shows, the laser sensor is mounted on the top of the robot and it does a 180 degree rotational scan, providing one reading per degree, whereas the eight sonar sensors are arranged in a half circle around it, providing only one reading per sensor for each scan. The sonar sensors incorporated by Pioneer 2 dx have a maximum range of 4 meters, while the SICK laser scanner can reach a maximum of 50 meters, but our approach defined the maximum in 8 meters.

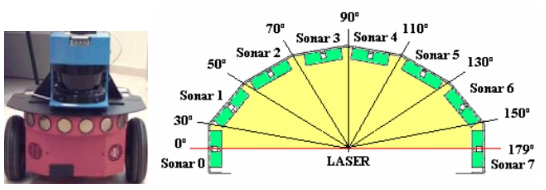
The software application used to carry out the experimentation of our approach was Player/Stage [c] as the network server for robot control, which provides a simple interface to the robot's sensors and actuators.

Finally, the testing scenarios were those available in our University building:

- Scenario I: Pioneer 2 dx located at the end of a corridor, which is closed with a glass window (see Fig. 5 (a)).
- Scenario II: Pioneer 2 dx located in front of the back doors, which also have glass windows (see Fig. 5 (b)).

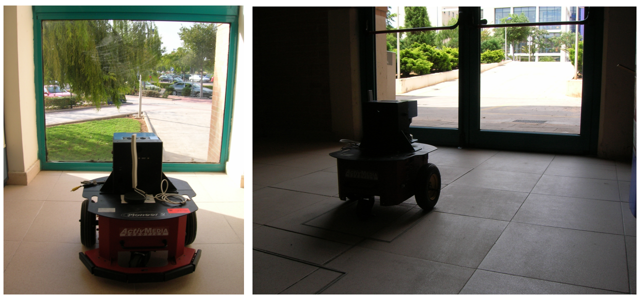
[a]http://www.mobilerobots.com
[b]http://www.sick.com
[c]http://playerstage.sourceforge.net

Fig. 4. Robotic platform used to test our approach.



(a) Pioneer 2 (b) Distribution of the sensor readings

Fig. 5. Scenarios to exemplify the results of our approach.



(a) Scenario I: rotbot at the end of a corridor (b) Scenario II: robot in front of the back doors

8.1. *Parametrization of our Approach*

For parametrizing our approach, the $FdSet$ is defined as:

$(FdSet, \mu_{FdSet})$ where,

$FdSet$ is a finite set $FdSet$ = {*at*, *very_close*, *close*, *quite_near*, *near*, *medium*, *quite_far*, *far*, *very_far*, *too_far*, *extremely_far*, *out_of_range*} and

$\mu_{FdSet} : FdSet \rightarrow [0,1] \in \Re$ defined by the triangular membership functions shown in Figure 6.

The membership functions of our $FdSet$ (μ_{FdSet}) have been defined by experimentation for indoor robot navigation: each limit depends on the diameter of the robot (d) (e.g. 60 cm for a Pioneer 2 dx mobile robot, 40 cm for an ERRATIC mobile robot [d], etc.). These divisions are motivated by the relation between the amount of distance in metres to an obstacle and the size of the object moving

[d]http://www.videredesign.com

towards it, for example: the same amount of distance in meters to an obstacle is considered cognitively closer by someone driving a truck than by someone riding a bike. However, other distance labels and limits could be established according to the application.

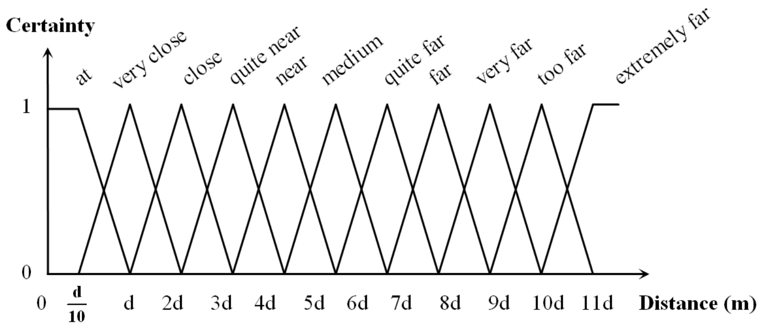


Fig. 6. Fuzzy distance Sets ($FdSets$) defined for our approach.

According to the previous $FdSet$ defined, the conceptual neighborhood relations explained in Section 4.2 and the general dissimilarity matrix defined in Table 2, a dissimilarity matrix between qualitative distance names ($QsimQd$) is obtained and shown in Table 4.

Table 4. Dissimilarity matrix for qualitative distances.

	at	*v.close*	*close*	*q.near*	*near*	*med.*	*q.far*	*far*	*v.far*	*t.far*	*ex.far*
at	0	1	2	3	4	5	6	7	8	9	10
v.close	-1	0	1	2	3	4	5	6	7	8	9
close	-2	-1	0	1	2	3	4	5	6	7	8
q.near	-3	-2	-1	0	1	2	3	4	5	6	7
near	-4	-3	-2	-1	0	1	2	3	4	5	6
med.	-5	-4	-3	-2	-1	0	1	2	3	4	5
q.far	-6	-5	-4	-3	-2	-1	0	1	2	3	4
far	-7	-6	-5	-4	-3	-2	-1	0	1	2	3
v.far	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2
t.far	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1
ex.far	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0

Moreover, the fuzzy set for robot speed, $FSpeed$, is defined by experts according to our application as:

$(FSpeed, \mu_{FSpeed})$ where,

$FSpeed$ is a finite set $FSpeed = \{$*stopped*, *very_slow*, *slow*, *quite_slow*, *medium*, *fast*, *very_fast*, *real_fast*$\}$ and

$\mu_{FSpeed} : FSpeed \rightarrow [0, 1] \in \Re$ defined by the triangular membership functions shown in Figure 7.

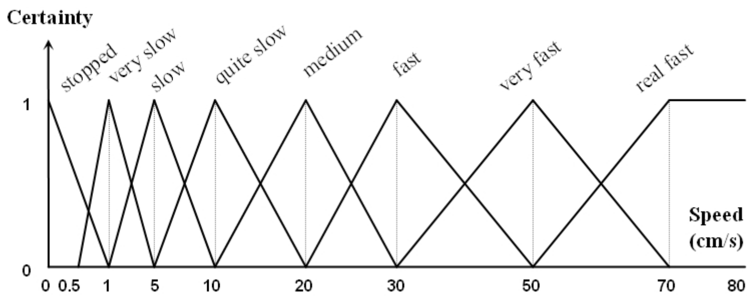


Fig. 7. Fuzzy sets defining the robot speed in centimeters per second

Table 6 shows the rules defined by our approach for relating the distances of our $FdSet$ with the speeds of our $FSpeed$ for obtaining a smooth robot speed according to the distance measured at the front or 90° and applying the defuzzification method presented in Eq.4.

Algorithm 6 Rules for defuzzification

```
if distance is extremely_far then speed is real_fast
if distance is too_far then speed is very_fast
if distance is very_far then speed is very_fast
if distance is far then speed is fast
if distance is quite_far then speed is fast
if distance is medium then speed is medium
if distance is near then speed is quite_slow
if distance is quite_near then speed is quite_slow
if distance is close then speed is slow
if distance is very_close then speed is very_slow
if distance is at then speed is stopped
```

8.2. *Tests and Results in Scenario I*

A video of the execution of our approach in **Scenario I** can be downloaded from our website [e]. The readings obtained by the sonar and laser sensors in this scenario are those shown in the Player Viewer screenshot in Figure 8 (b) and the FDPs obtained are the following ones, respectively:

$FDP(sonar)$=
[[0, 29], [(at, 0.11), (very_close, 0.89)]],
[[30, 49], [(near, 0.8), (medium, 0.2)]],
[[50, 69], [(very_close, 0.04), (close, 0.96)]],
[[70, 149], [(close, 0.66), (quite_near, 0.35)]],
[[150, 179], [(quite_near, 1)]].

$FDP(laser)$=
[[0, 53], [(very_close, 0.6), (close, 0.4)]],
[[54, 71], [(close,0.69), (quite_near, 0.31)]],
[[72, 109], [(extremely_far, 1)]],
[[110, 159], [(close, 0.6), (quite_near, 0.4)]],
[[160, 179], [(quite_near, 0.68), (near, 0.32)]].

After determining a dissimilarity threshold by experimentation ($|T| > 2.5$) and comparing the obtained sonar and laser fuzzy distance patterns ($FDP(sonar)$ and $FDP(laser)$), two large dissimilarities are found (-2.8 and 7.73, respectively). An example of how to obtain the corresponding DF between the corresponding $FdSet$s from Eq. 2 is shown next:

$DF(FdSet(sonar), FdSet(laser))$ =
DF([(near, 0.8),(medium, 0.2)],[(very_close, 0.6),(close, 0.4)]) =
$= dSimQd(near, very\_close) \cdot (0.8 \cdot 0.6) + dSimQd(near, close) \cdot (0.8 \cdot 0.4) + dSimQd(medium, very\_close) \cdot (0.2 \cdot 0.6) + dSimQd(medium, close) \cdot (0.2 \cdot 0.4) = (-3) \cdot 0.48 + (-2) \cdot 0.32 + (-4) \cdot 0.12 + (-3) \cdot 0.08 = -2.8$

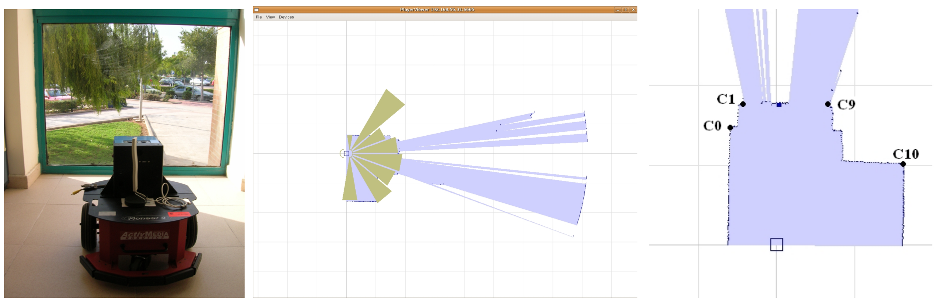
These dissimilarities are interpreted by our approach as a *Sound Reflection* in the angular positions 30-49 and a *Glass or mirror* in the angular positions 72-109. The final FDP obtained is the following one:

$FDP(final)$=
[[0, 29], [(very_close,0.6), (close,0.4)], Simple Obstacle],
[[30, 49], [(very_close,0.6), (close,0.4)], Sound Reflection], $DF = -2.8$
[[50, 53], [(very_close,0.6), (close,0.4)], Simple Obstacle],
[[54, 71], [(close, 0.69), (quite_near,0.31)], Simple Obstacle],

[e] http://www.c-robots.com/IJUFKS/scenario1.MOV

[[72, 109], [(close,0.66), (quite_near,0.35)], Glass or mirror], $DF = 7.73$
[[110, 159], [(close, 0.6), (quite_near,0.4)], Simple Obstacle],
[[160, 179], [(quite_near,0.68), (near,0.32)], Simple Obstacle].

Fig. 8. Results obtained from the tests in Scenario I.



(a) Scenario I (b) Player Viewer application showing laser and sonar readings in Scenario I (c) Corners that build *RS*s in Scenario I

The *FdSet* of the final *FDP* used to obtain the robot speed is that corresponding to the front of the robot or 90°: [(close, 0.66),(quite_near,0.35)]. Therefore, according to Eq. 4, the speed obtained is: 0.08 cm/s.

There are no large discontinuities of distance in the final *FDP* (the *DF*s obtained are: 0, 0.91, 0.04, 0.05 and 0.93) so all the *RS*s built from the corners obtained using Peris and Escrig's approach[18] approach are closed:

Corners[(47,Concave),(72,Convex)] → RS(0,1): Closed
Corners[(72,Convex),(75,Convex)]→ RS(1,2): Closed
Corners[(75,Convex),(78,Convex)]→ RS(2,3): Closed
Corners[(78,Convex),(79,Convex)]→ RS(3,4): Closed
Corners[(79,Convex),(83,Convex)]→ RS(4,5): Closed
Corners[(83,Convex),(84,Convex)]→ RS(5,6): Closed
Corners[(84,Convex),(87,Convex)]→ RS(6,7): Closed
Corners[(87,Convex),(95,Convex)]→ RS(7,8): Closed
Corners[(95,Convex),(111,Convex)]→ RS(8,9): Closed
Corners[(111,Convex),(148,Concave)]→ RS(9,10): Closed

It is important to note that the *Sound Reflection* detected in the final *FDP* coincides with the angular position of the first corner: 47, as it is shown in Figure 8. It is also interesting to note that both the angular positions that are classified as *Glass or mirror* coincide with *convex* corners, as shown in Figure 8 (c).

As the *Glass or mirror* detected is situated between the angular positions 72 and 109, the following *RS*s are discarded: RS(1,2), RS(2,3), RS(3,4), RS(4,5), RS(5,6), RS(6,7), RS(7,8) and RS(8,9). Therefore, the *RS*s considered by the robot are RS(0,1) and RS(9,10), and RS(1,9) is built as result of the integration of the final *FDP* with the *CornersVector*. The angular position and type of these corners correspond to reality, as can be seen by comparing them with the image in Figure 8(c):

Corners[(47,Concave),(72,Convex)] → RS(0,1): Closed
Corners[(72,Convex), (111,Convex)] → New RS(1,9): Closed
Corners[(111,Convex),(148,Concave)]→ RS(9,10): Closed

In Figure 8(c), it can be seen that a corner has been missed. This it is not a consequence of discarding *RS*s by our approach, because the corners are detected using Peris and Escrig's approach [18].

8.3. *Tests and Results in Scenario II*

A video of the execution of our approach in **Scenario II** can be downloaded from our website [f]. The readings obtained by the sonar and laser sensors in this scenario are those shown in Player Viewer screenshot in Fig. 9(b) and the *FDP*s obtained are the following ones, respectively:

FDP(*sonar*)=
[[0, 29], [(too_far, 0.18), (extremely_far, 0.82)]],
[[30, 49], [(medium, 0.68), (quite_far, 0.32)]],
[[50, 129], [(quite_near, 0.73), (near, 0.28)]],
[[130, 149], [(near, 0.84), (medium, 0.16)]],
[[150, 179], [(close, 0.14), (quite_near, 0.86)]].

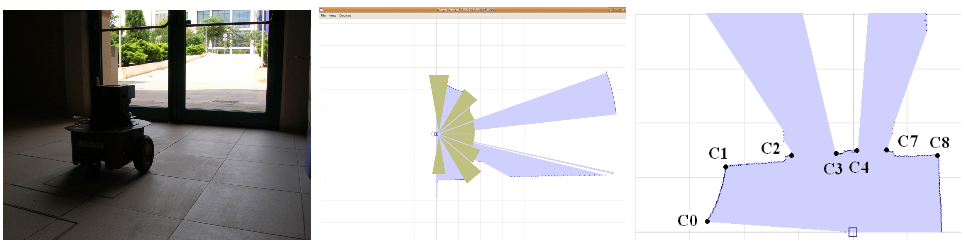
FDP(*laser*)=
[[0, 34], [(medium, 0.69), (quite_far, 0.31)]],
[[35, 58], [(near, 0.5), (medium, 0.5)]],
[[59, 78], [(extremely_far, 1)]],
[[79, 92], [(quite_near, 0.75), (near, 0.25)]],
[[93, 109], [(extremely_far, 1)]],
[[110, 129], [(quite_near,0.37), (near, 0.63)]],
[[130, 143], [(near, 0.79), (medium, 0.21)]],
[[144, 179], [(quite_near, 0.51), (near, 0.49)]].

By comparing the sonar and laser *FDP*s, two large dissimilarities are found (-4.51, 6.69 and 6.69, respectively), which are interpreted by our approach as a *Sound Reflection* in the angular positions 0-29 and a *Glass or mirror* in the angular positions 59-78 and 93-109. The final *FDP* obtained as follows:

[f] http://www.c-robots.com/IJUFKS/scenario2.MOV

$FDP(final)$=
[[0, 29], [(medium, 0.69), (quite_far,0.31)], Sound Reflection], $DF = -4.51$
[[30, 34], [(medium, 0.69), (quite_far, 0.31)], Simple Obstacle],
[[35, 58], [(near,0.5), (medium,0.5)], Simple Obstacle],
[[59, 78], [(quite_near,0.73), (near,0.28)], Glass or mirror], $DF = 6.69$
[[79, 92], [(quite_near,0.75), (near,0.25)], Simple Obstacle],
[[93, 109], [(quite_near,0.73), (near,0.28)], Glass or mirror], $DF = 6.69$
[[110, 129], [(quite_near,0.37), (near,0.63)], Simple Obstacle],
[[130, 143], [(near, 0.79), (medium, 0.21)], Simple Obstacle],
[[144, 179], [(quite_near, 0.51), (near, 0.49)], Simple Obstacle].

Fig. 9. Corner detection and *RS*s building in Scenario II.



(a) Scenario II (b) Laser and sonar readings (c) Corners that build *RS*s in Scenario II

The *FdSet* of the final *FDP* used to obtain the robot speed is that corresponding to the front of the robot or 90°: [(quite_near, 0.75), (near, 0.25)]. Therefore, according to Eq. 4, the speed obtained is: 0.12 cm/s.

There are no large discontinuities of distance in the final *FDP* (the *DF*s obtained are: −0.81, −1.24, −0.03, 0.03, 0.36, 0.58 and −0.72) therefore, all the *RS*s built from the corners obtained using the Peris and Escrig's approach [18] are closed:

Corners[(3,Convex),(32,Concave)] → RS(0,1): Closed
Corners[(32,Concave),(59,Convex)] → RS(1,2): Closed
Corners[(59,Convex),(79,Convex)] → RS(2,3): Closed
Corners[(79,Convex),(92,Convex)] → RS(3,4): Closed
Corners[(92,Convex),(93,Convex)] → RS(4,5): Closed
Corners[(93,Convex),(94,Convex)] → RS(5,6): Closed
Corners[(94,Convex),(110,Convex)] → RS(6,7): Closed
Corners[(110,Convex),(132,Concave)] → RS(7,8): Closed

It is important to note that the *Sound Reflection* detected in the final *FDP* coincides with the left round wall, as shown in Figure 9 (c). It is also interesting to note that the angular positions of the *Glass or mirror* both coincide with *convex* corners, as shown in Figure 9 (c).

As the *Glass or mirror* detected is situated between the angular locations 72-109 and 93-109, the following *RS*s are discarded: RS(2,3), RS(3,4) and RS(7,8). Therefore, the *RS*s considered by the robot are RS(0,1), RS(1,2), R(3,4) and RS(9,10) and the angular location and type of these corners correspond to reality as can be seen when comparing them with the image in Figure 9 (c):

Corners[(3,Convex),(32,Concave)] → RS(0,1): Closed
Corners[(32,Concave),(59,Convex)] → RS(1,2): Closed
Corners[(59,Convex),(79,Convex)] → RS(2,3): Closed
Corners[(79,Convex),(92,Convex)] → RS(3,4): Closed
Corners[(92,Convex),(110,Convex)] → New RS(4,7): Closed
Corners[(110,Convex),(132,Concave)] → RS(7,8): Closed

8.4. *Summary of the Results*

As a summary, our approach has enabled the robot to succeed in:

(i) detecting mirrors and glass in the robot world,
(ii) obtaining the real distance to corners, since the sound reflections are detected,
(iii) detecting non-working sensors, such as laser sensor disconnection, and avoiding crashing into obstacles when lacking the information provided by them,
(iv) approaching obstacles (including glass) at a smooth speed and avoiding crashing into them,
(v) properly classifying reference systems (*RS*s) as *open* or *closed*, including those that incorporate glass surfaces,
(vi) recognizing the features of the environment that it is facing, according to the *RS*s detected, as they are characterized in Algorithm 4.

9. Conclusion

An approach to distance sensor data integration that provides a robust interpretation of the robot environment has been presented in this paper. Our approach consists in obtaining patterns of fuzzy distance zones from sonar and laser sensor readings; comparing these patterns in order to detect non-working sensors; and integrating the patterns obtained to detect obstacles of any sort. A dissimilarity factor (DF) between fuzzy sets has been defined and applied to this approach. And a method for defuzzifying the obtained fuzzy distances into a fuzzy robot speed has been also used.

In order to test our approach, an ActivMedia Pioneer 2 dx mobile robot incorporating a SICK LMS-200 laser range scanner and the Player/Stage control interface have been used. However, our approach is extensible to other types of distance sensors (such as infrared sensors) and other kinds of mobile robots containing distance sensors.

The more important results obtained show that this approach enables the robot to: detect non-working sensors and special obstacles such as mirrors and glass windows in the robot world; approach obstacles (including glass surfaces) at a smooth speed and avoid crashing into them; and properly classify reference systems (*RS*s) as *open* or *closed*, including those with glass surfaces.

As future work, we intend to: (i) design an ontology to give meaning to all the qualitaive concepts extracted from the robot environment; (ii) extend our sensor data integration approach in order to include a camera; (iii) design an approach to integrate visual information obtained by the robot camera with the distance information provided by our approach so that a notion of depth could be included in the description of visual landmarks of the robot world.

Acknowledgements

This work has been partially supported by the Generalitat Valenciana (BFPI06/219). We would like to acknowledge the reviewers for their comments and Pablo Ródenas, Jorge Grande and the *RobInLab - Robotics Intelligent Lab* at University Jaume I for their technical support.

References

1. J. Hawkins, S. Blakeslee, "On Intelligence", Times Books, 2004.
2. Honghai Liu, (2008). "A Fuzzy Qualitative Framework for Connecting Robot Qualitative and Quantitative Representations". *IEEE Transactions on Fuzzy Systems*, 16(6):1522–1530.
3. Martin, C., Schaffernicht, E., Scheidig, A., and Gross, H.-M. (2006). "Multi-modal sensor fusion using a probabilistic aggregation scheme for people detection and tracking". *Robotics and Autonomous Systems*, 54(9):721–728.
4. Lai, X.-C., Kong, C.-Y., Ge, S. S., and Mamun, A. A. (2005). "Online map building for autonomous mobile robots by fusing laser and sonar data". In *IEEE International Conference on Mechatronics and Automation*, volume 2, pages 993–998.
5. Li, X., Huang, X., Dezert, J., Duan, L., and Wang, M. (2005). "Robot map building from sonar and laser information using DSmT with discounting theory". *International Journal of Information Technology*, 3(2):78–85.
6. Diosi, A., Taylor, G., and Kleeman, L. (2005). "Interactive SLAM using laser and advanced sonar". In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1103–1108.
7. Diosi, A. and Kleeman, L. (2004). "Advanced sonar and laser range finder fusion for simultaneous localization and mapping". In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, volume 2, pages 1854–1859.
8. Vamossy, Z., Kladek, D., and Fazekas, L. (2004). "Environment mapping with laser-based and other sensors". In *Proceedings of the International Workshop on Robot Sensing (ROSE)*, pages 74–48.
9. Herrero, D., Zamora, M., and Martinez, H. (2002). "Fusion de sonar y laser mediante

mapas de segmentos difusos". In *Proceedings of the III Workshop en Agentes Fsicos (WAF), Spain*, pages 57–69. Springer-Verlag.

10. Reece, S. and Durrant-Whyte, H. F. (1995). "A qualitative approach to sensor data fusion for mobile robot navigation". In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 36–41.
11. Reece, S. (1997a). "Data fusion and parameter estimation using qualitative models: The qualitative kalman filter". In *Proceedings of the Eleventh International Qualitative Reasoning Workshop (QR)*, pages 143–153.
12. Reece, S. (1997b). "Qualitative model-based multisensor data fusion and parameter estimation using infinity-norm dempster-shafer evidential reasoning". In *SPIE Eleventh Annual International Symposium on Aerospace/Defence Sensing, Simulation, and Controls*, pages 52–63.
13. Reece, S. (2000). "Self-adaptive multi-sensor systems". In P. Robertson, H. S. and Laddaga, R., editors, *First International Workshop Self-Adaptive Software (IWSAS)*, volume 1936 of *Lecture Notes in Computer Science (LNCS)*, pages 224–241. Springer-Verlag.
14. Zivkovic, Z., Booij, O., Kröse, B. J. A., Topp, E. A., and Christensen, H. I. (2008). "From sensors to human spatial concepts: An annotated data set". *IEEE Transactions on Robotics*, 24(2):501–505.
15. Zender, H., Mozos, Ó. M., Jensfelt, P., Kruijff, G.-J. M., and Burgard, W. (2008). "Conceptual spatial representations for indoor mobile robots". *Robotics and Autonomous Systems*, 56(6):493–502.
16. Oliveira, L., Costa, A., Schnitman, L., and de Souza, J. A. M. F. (2005). "An architecture of sensor fusion for spatial location of objects in mobile robotics". In Bento, C., Cardoso, A., and Dias, G., editors, *Progress in Artificial Intelligence, 12th Portuguese Conference on Artificial Intelligence, EPIA 2005, Covilhã, Portugal, December 5-8, 2005, Proceedings*, volume 3808 of *Lecture Notes in Computer Science*, pages 462–473. Springer.
17. Falomir, Z., Jimenez-Ruiz, E., Museros, L. and Escrig, M. T. (2010). "Describing Images using Qualitative Models and Description Logics". *Journal of Spatial Cognition and Computation*, to appear in december 2010.
18. Peris, J. C. and Escrig, M. T. (2005). "Cognitive maps for mobile robot navigation: A hybrid representation using reference systems". In *Proceedings of the 19th International Workshop on Qualitative Reasoning (QR'05)*, Granz, Austria.
19. Zadeh, L. (1965). "Fuzzy sets". *Information and Control*, 8:338–353.
20. Allen, J. (1981). "An Interval-Based Representation of Temporal Knowledge". In *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*, pages 221–226.
21. Freksa, C. (1991). "Qualitative spatial reasoning". In Mark, D. M. and Frank, A. U., editors, *Cognitive and Linguistic Aspects of Geographic Space*, NATO Advanced Studies Institute, pages 361–372. Kluwer, Dordrecht.
22. J. Galindo, "Conjuntos y Sistemas Difusos (Lógica Difusa y Aplicaciones)", Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, 2007.
23. Freksa, C. (1992). "Using orientation information for qualitative spatial reasoning". In Frank, A. U., Campari, I., and Formentini, U., editors, *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space. Intl. Conf. GIS—From Space to Territory*, volume 639 of *Lecture Notes in Computer Science*, pages 162–178, Berlin. Springer.