International Journal of Approximate Reasoning 51 (2010) 912–932

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

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

Fuzzy uncertainty modeling for grid based localization of mobile robots

D. Herrero-Pérez [a,*], H. Martínez-Barberá [a], K. LeBlanc [b], A. Saffiotti [b]

[a] Department of Information and Communications Engineering, University of Murcia, 30100 Murcia, Spain
[b] Center for Applied Autonomous Sensor Systems, Department of Technology, Örebro University, 70182 Örebro, Sweden

ARTICLE INFO

Article history:
Received 22 December 2009
Received in revised form 18 June 2010
Accepted 21 June 2010
Available online 30 June 2010

Keywords:
Fuzzy logic
Fuzzy mathematical morphology
Robot localization
Spatial uncertainty
State estimation
RoboCup

ABSTRACT

This paper presents a localization method using fuzzy logic to represent the different facets of uncertainty present in sensor data. Our method follows the typical predict-update cycle of recursive state estimators to estimate the robot's location. The method is implemented on a fuzzy position grid, and several simplifications are introduced to reduce computational complexity. The main advantages of this fuzzy logic method compared to most current ones are: (i) only an approximate sensor model is required, (ii) several facets of location uncertainty can be represented, and (iii) ambiguities in the sensor information are directly represented, thus avoiding having to solve the data association problem separately. Our method has been validated experimentally on two different platforms, a legged robot equipped with vision and a wheeled robot equipped with range sensors. The experiments show that our method can solve both the tracking and the global localization problem. They also show that this method can successfully cope with ambiguous observations, when several features may be associated to the same observation, and with robot kidnapping situations. Additional experiments are presented that compare our approach with a state-of-the-art probabilistic method.

1. Introduction

Uncertainty is a term used in diverse scientific fields, including philosophy, statistics, economics, finance, engineering and science. Depending on the field, uncertainty may refer to predictions of future events, to physical measurements already made, or to the unknown. When this term is used in the field of *Robotics*, it usually refers to noisy measurements which are represented in terms of a range of values which are expected to enclose the actual value of the observed variable. From a probabilistic point of view, this uncertainty is usually modeled using a probability distribution, whose parameters are identified by repeating the measure under the same conditions a large number of times. In other words, probabilistic functions should be experimentally calculated, which requires the ability to reproduce the measure in similar conditions. Unfortunately, reproducing the factors that influence a measurement is not always possible, because these factors are often unknown.

In the field of mobile robotics, the *localization problem* consists of answering the question *"Where am I?"* from the robot's point of view [1], given a representation of the robot's world and measurements from the robot's sensors. Measurements may refer to the robot's perception of its own motion (sometimes called proprioception), or to the robot's perception of its surrounding environment (sometimes called exteroception). Unfortunately, when using real physical sensors, these measurements are commonly affected by uncertainty of different types, including vagueness, imprecision, ambiguity, unreliability, and random noise. Measurements may also be affected by several simultaneous factors, which are not necessarily

* Corresponding author.
E-mail address: dherrero@um.es (D. Herrero-Pérez).

0888-613X/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2010.06.001

independent. Because of this, it is important that the uncertainty in the measurements is carefully represented in a robotic system. In particular, the used uncertainty representation formalism should be able to capture the different types of uncertainty and to account for the differences between them. Fuzzy logic [2] has been shown to provide expressive tools and techniques to represent and handle the different facets of the uncertainty in the measurements [3]. Techniques based on fuzzy logic have already been successfully used in several domains related to robotics, such as fuzzy control [4,5], fuzzy modeling [6,7], fuzzy geometric reasoning [8,9], and fuzzy information fusion [10,11].

In this paper, we show how fuzzy logic can be used to build an integrated framework that addresses the robot localization problem in an effective way. This framework takes inspiration from early ideas proposed by Buschka [12] and Herrero [13], and generalizes them into a comprehensive framework for fuzzy robot localization. The key move to the realization of this framework has been to combine two general schemas, respectively coming from the tradition of fuzzy systems and the tradition of position estimation:

- the entire framework follows the typical *Fuzzification–Fuzzy inference–Defuzzification* schema of most fuzzy systems;
- the fuzzy inference part follows the typical *Predict–Update* cycle of recursive state estimators.

The method proposed in this paper uses fuzzy logic to account for errors, imprecision and ambiguity in the perceptual information, as well as for the uncertainty in the estimates of the robot's motion. An important advantage of the ability to handle ambiguity is that our method does not need to rely on a separate data association step to uniquely associate sensor readings to environmental features. This helps to increase the robustness of our method, since data association is one of the most common reasons for failures in robot localization.

The method presented in this paper has been experimentally validated on several different platforms. In this paper, we show representative experiments on two different testbeds: a legged robot in the RoboCup environment, and a wheeled robot in an indoor office environment. The two testbeds use different types of sensors: color vision in the first case, and a laser rangefinder in the second one. We demonstrate the ability of our method to solve the robot localization problem both during normal robot navigation and in pathological situations where the robot is manually picked-up and moved to a new location – the so-called "kidnapped robot problem". We also demonstrate the ability of our method to correctly handle ambiguity, by showing experiments in which multiple indistinguishable features may generate the same observations.

The rest of this paper is structured as follows: Section 2 discusses the relevant related literature. Section 3 presents the fuzzy framework for robot localization in general terms. Section 4 presents a concrete instance of this framework based on the use of fuzzy position grids. The robotic platforms and the environments used to experimentally validate our method are introduced in section 5. Sections 6 and 7 present the experimental results obtained on the legged and on the wheeled platforms, respectively. Finally, conclusions are presented in Section 8.

2. Related work

The localization methods in the field of mobile robotics are usually classified according to the type of problem they attack [14], the most common distinction being the one between *tracking* or *local* techniques, and *global* techniques. Global localization methods address the problem without *a priori* knowledge of the initial robot's location, i.e., the robot is placed somewhere in the environment and has to localize itself from scratch. Tracking localization methods aim to compensate small dead-reckoning errors under the assumption that the initial position is known. Commonly, local techniques present advantages in accuracy and efficiency, while global techniques are much more robust [15], mainly because they are able to recover from failures. Therefore, these techniques are sometimes combined; the process usually consists of combining a technique that solves the *global localization* problem with other that solves the *local localization* problem with more accuracy and less computational cost [16–18].

In order to address the localization problem, robots usually have access to relative (also called dead-reckoning) and absolute (reference-based systems) position measurements [19]. These measurements are provided by different sensors which give the feedback about the actions performed by the robot and the observations around its pose. What makes this problem difficult is the presence of uncertainty in the position measurements, thus uncertainty representation is a key point to solve it. The most used approaches are based on probabilistic techniques, since these provide powerful statistical tools to manage noisy measurements. These techniques usually rely on probabilistic models of the measurements that take a frequentist interpretation of the underlying noise. The accuracy of the estimates produced depends on the how well these models reflect the real noise affecting the measurements. Moreover, these models often assume that errors are normally distributed and that they are small enough to allow linear approximations. Unfortunately, practical experience suggests that these assumptions are often violated in reality for robotic systems.

Another, arguably less developed family of approaches makes use of fuzzy sets to represent the different facets of uncertainty that affect to the measurements. Fuzzy techniques also provide a wide set of tools to combine those measurements, and to address matching problems based on the similarity interpretation of fuzzy logic [20]. Moreover, fuzzy logic based techniques may be applicable in domains where the assumptions of other methods are not satisfied: for instance, they have often been used in cases when a stochastic sensor model cannot be easily elicited [21]. These arguments have induced to several authors to make use of fuzzy sets as uncertainty representation of locations in diverse problems and applications

of robotics instead of more popular choice of using probabilities. Some examples are mapping, localization, multi-sensor and multi-robot sensor fusion, and Simultaneous Localization and Map Building (SLAM) problems.

Some authors have approached the map building problem building models of the world (maps) based on fuzzy logic, such as fuzzy segment maps [22] or fuzzy grid maps [21]. The former approach represents the location uncertainty of the features, line-segments, by fuzzy sets that incorporate different sources of uncertainty, including dead-reckoning, sensor noise, and clustering errors. The features are then combined by fuzzy matching, grounded on the similarity interpretation of fuzzy logic. The latter approach aggregates the sensor data over time into two fuzzy grid representations of the world, one representing the empty (traversable) space and one representing the occupied (non-traversable) space. Data aggregation is performed using fuzzy operators.

Fuzzy logic has also been used to address the localization problem, namely using some form of fuzzy matching [23,24], or combining position measurement information (landmark-based) using fuzzy aggregation operators [25]. The former approach defines the localization problem as the problem of finding the best correspondence between a local map obtained from observations using fuzzy metric features, and a fuzzy model of the environment. The latter approach treats each observation as a source of partial location information for combining relative and absolute position measurements. The method proposed in [12] combines ideas from the Markov localization approach proposed in [26] with ideas from the fuzzy landmark-based approach. Fuzzy logic has also been used to improve the accuracy and robustness of other localization methods; in particular controlling some aspects of the localization method or switching between localization methods with different properties. For instance, on-line adaptation of the gain of an Extended Kalman Filter, which weights the importance of absolute and relative position measurements, it using fuzzy membership functions for improving the accuracy and robustness of the localization method [27,28]. In order to make use of the advantages of probabilistic Markov Localization [14] and Monte Carlo Localization [29] methods, fuzzy membership functions are used to model the uncertainty of both methods using different heuristics to enable the adaptation of the resulting method [30–32] to different levels of noise and other specific properties of such a methods, like sparsity.

A related problem that has been addressed using fuzzy logic is the problem of fusing location information in a multi-robot system. In [33,34], the object's position is represented in a fuzzy grid taking into account both the uncertainty in the observations and in the self-localization of the observing robot, which is propagated to the object-position estimation. Fuzzy grids corresponding to the observations from different robots are fused using fuzzy operators into a combined fuzzy grid, which represents the consensus among the robots. On the other hand, methods based on Bayesian approaches [35,36] aim to estimate a compromise between the different sources of information, and thus they can incur well-known problems when information is unreliable [33]. For example, most current Bayesian based techniques average information from different robots in order to obtain a trade-off, which generates uncertain estimations when unreliable measurements.

Finally, a few works exist which have addressed the Simultaneous Localization and Mapping (SLAM) problem using techniques based on fuzzy logic. For example, Gasós and colleagues [37,38] propose to extract metric features from range data, thus building a local representation, and then to match them to a global representation of the world. The matching process results in a new estimate of the robot's position, which indicates either the area of the local representation was previously explored or not. In the former case, the local representation is added to the global one; in the latter, the global representation is updated with the local one. Other examples [39] make use of fuzzy logic for inferring the uncertainty in robot's location after motion commands, and then matching process is performed using genetic algorithms, which search the most probable map given the location information. The correspondence problem is solved exploiting the property of natural selection, which supports better performing individuals to survive in the competition.

3. Overall localization framework

In our approach, we consider the robot self-localization problem as a *fuzzy estimation problem*. At any point in time during navigation, the robot has a belief about where it is based on the information collected so far. This belief is not represented as a single location, but as a fuzzy set over the space of possible locations: each location has a degree of membership that reflects how much this location could be the actual one. The localization problem consists of maintaining this belief as the robot moves and collects new noisy measurements.

Burgard and colleagues [26] proposed a similar approach to the robot localization problem based on a probabilistic framework. In their approach, called *Markov Localization*, the robot's belief is represented by a probability density over the space of possible locations. This approach constitutes the basis of several currently used localization methods; it combines information from multiple sensors in the form of relative and absolute measurements to form a combined probabilistic belief in the robot's location. Although Markov localization is stated as a Bayesian estimation problem, many ideas and assumptions are suitable and can also be exploited in a fuzzy framework. In particular, the *Markov assumption*, which states that given knowledge of the current state, the future is independent of the past, and vice versa. In other words, once the previous robot's location x_{t-1} is known, it does not matter how this location was reached or what has been sensed.

In the rest of this section, we describe the main components of our fuzzy localization framework. We first present the notion of *fuzzy locations*, our main device to represent uncertain locational information, and how these are used to represent the robot's belief about its own location. We then discuss the way to model the effect of actions and of new observations over

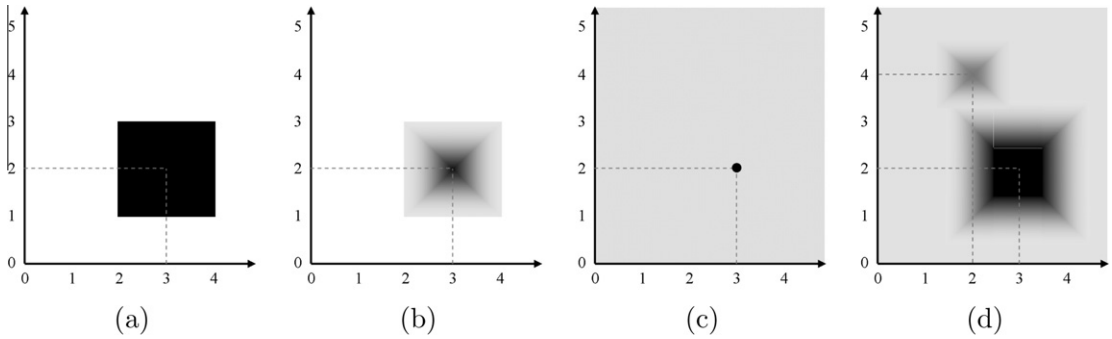


Fig. 1. Four examples of fuzzy locations in a 2D space that illustrate different types of spatial uncertainty: (a) imprecision, (b) vagueness, (c) unreliability, and (d) combined. Grey levels indicate membership values, with white representing 0 and black representing 1.

the robot's belief. Finally, we put the above components together and describe the top-level logic of our fuzzy self-localization process.

3.1. Fuzzy locations

In this paper, we deal with location information of objects (mainly, of a robot) in a given set X of all possible spatial locations. For instance, X can be a six-dimensional space encoding the position coordinates (x,y,z) of the object together with its (θ,ϕ,η) orientation angles.

In order to represent uncertain information about the location of objects, we introduce the notion of *fuzzy locations*. A fuzzy location for a given object is a fuzzy subset $\mu: X \to [0,1]$ of the spatial set X.[1] We adopt a possibilistic interpretation of the membership values [40]: for any $x \in X$, the value of $\mu(x)$ is read as the degree of possibility that the target object is located at x, given the available information. Total ignorance is represented by the fuzzy location $\mu(x) = 1$ for all $x \in X$.

Fuzzy locations allow us to represent different types of locational uncertainty. As an illustration, consider the four examples of fuzzy locations shown in Fig. 1, cast in a two-dimensional space (X,Y) for graphical simplicity. Fuzzy location (a) represents the item of *imprecise* information "the object is at position $(3 \pm 1{,}2 \pm 1)$". Imprecision may, for instance, originate from the limited resolution of a sensor; in the shown example, all the positions in the dark square are equally and fully possible, but positions outside it are impossible. Fuzzy location (b) is a way to represent the item of *vague* information "the object is approximately at position $(3,2)$". Vagueness may, for instance, originate from a linguistic input; in the shown example, position $(3,2)$ is fully possible, and the degree of possibility of other positions decreases with their (Manhattan) distance from $(3,2)$. Fuzzy location (c) represents the item of *unreliable* information "the object may be at position $(3,2)$", where unreliability may originate in uncertainty in data association, e.g., the measurement could actually refer to a different object. In the shown example, position $(3,2)$ is fully possible, but all other positions are also possible to a small degree, as indicated by the uniform gray background. This corresponds to the fact that, should the information be wrong, the object can be located just anywhere. Finally, fuzzy location (d) represents information affected by all of the above types of uncertainty, including ambiguity. Here, the positions in the square $(3 \pm 1{,}2 \pm 1)$ are fully possible, and positions in the proximity of it are partly possible; there is also a mild possibility for positions around $(2,4)$, e.g., due to a second, weaker observation; finally, a small uniform degree of possibility is assigned to all other positions, reflecting the partial unreliability of this fuzzy location – e.g., the information may be out of date.

3.2. Robot's belief

At any point in time t, the robot maintains a belief G_t about its own location. In this work, we represent this belief by a fuzzy location $G_t: X \to [0,1]$ over the n-dimensional set X of all possible locations. Thus, the fuzzy self-localization problem is defined as the problem of maintaining G_t in time, given the motion of the robot and the observations from the robot's sensors. Intuitively, the goal of self-localization is to keep this belief as close as possible to the real location of the robot. In the ideal case, G_t is a distribution with a single peak of value 1 at the real robot location, and zero values everywhere else. In practice, because of noise and errors in sensing and acting, G_t will always be affected by uncertainty.

3.3. Fuzzy action model

Our fuzzy localization framework follows the typical predict-update cycle of recursive state estimators: at every cycle, the next belief state is predicted from the previous one using information on how the robot has moved; then, this prediction is

[1] For the sake of notational simplicity, in this paper we shall identify fuzzy subsets A of X with their membership functions $\mu_A: X \to [0,1]$.

updated using the new observations from the sensors. In our framework, both the robot's belief, the information about the robot's motion, and the information about new observations are all modeled as fuzzy sets. Correspondingly, the prediction and update functions are defined through operations on fuzzy sets.

In order to predict the belief state G_t, we rely on operators from fuzzy mathematical morphology [41]. Fuzzy mathematical morphology extends traditional mathematical morphology [42,43] to fuzzy sets, and it provides a wide range of well-founded tools to deal with fuzzy objects and fuzzy relations in digital images. By seeing fuzzy locations on the set X as fuzzy objects in an image space, we can use these tools to process fuzzy locations [44]. In particular, we can use fuzzy dilation to account for the transformation of the belief state G_t after a robot's motion.

Fuzzy dilation is an operator that performs a global transformation of a fuzzy subset G of X, by repeatedly applying a local transformation around each element $x \in X$. The transformation is defined by a fuzzy "structuring element" B, itself a fuzzy subset of X. Formally, the fuzzy dilation of G by B, denoted $G \oplus B$ is defined by[2]:

$$(G \oplus B)(x) = \sup_{y \in X} T(G(x), B(y - x)), \tag{1}$$

where T is a triangular norm, or t-norm [45], expressing dilation operation as a degree of intersection [46]. Intuitively, the result of a fuzzy dilation is a fuzzy distribution which is spatially expanded from G, where B determines the shape of the expansion. If B is isotropic, then G is expanded in all directions; if the center of mass of B is displaced with respect to its origin, then G is expanded more in the direction of the displacement; if B has fuzzy contours, then the contours of G are blurred; and so on.

In our approach, we model the motion of the robot from time $t-1$ to time t by a structuring element B_t. We read the value of $B_t(x)$ as the degree of possibility that the robot has moved to position x at time t, given that it was at the origin of B at time $t-1$. In other words, B_t encodes our fuzzy *action model*. Given the robot's belief G_{t-1} at time $t-1$, then, we compute the robot's *predicted belief* at time t, denoted by G'_t, by

$$G'_t = G_{t-1} \oplus B_t. \tag{2}$$

Typically the action model B_t is given by the composition of three components: a translational component, a rotational component, and a blurring component. The first two components are computed from the robot's odometric data, and their effect is to displace the values in the G_{t-1} distribution from its center to account for the measured motion. The third component accounts for the uncertainty introduced by the noise in the odometric measures, and its effect is to blur the G_{t-1} distribution. In practice, these three components can be represented by three separate structuring elements B^{ρ}_t, B^{θ}_t, B^{ϵ}_t: the properties of fuzzy dilation guarantee that the result obtained by applying these three elements one after the other is the same as applying the combined element [44].

3.4. Fuzzy sensor models

The observations that provide location information are also represented in our framework using fuzzy locations. Given a sensor observation r obtained from sensor i at time t, we let $S^i_t(\cdot|r) : X \to [0,1]$ be a fuzzy location that represents the information provided by this observation about the robot's location. Thus, for any n-dimensional location $x \in X$, $S^i_t(x|r)$ represents the degree of possibility that the robot is at x given that sensor i has made observation r.

We call S^i_t the fuzzy *sensor model* of sensor i. Note that S^i_t is an inverse model, since it represents the belief of being at location x given an observation, and not vice versa. The way S^i_t is defined depends on the specific sensor and the specific type of feature observed: if the same sensor can observe multiple types of features, multiple sensor models are required for it. In the next sections, we shall show examples of the fuzzy sensor models defined in our domain.

All the observations $\{r_1, \ldots, r_n\}$ made at time t are used in our fuzzy estimation cycle to update the predicted belief G'_t. More precisely, we combine G'_t with the fuzzy locations $S^i_t(\cdot|r_i),\ i = 1, \ldots, n$ as follows:

$$G_t(\cdot) = G'_t(\cdot) \cap S^i_t(\cdot|r_1) \cap S^i_t(\cdot|r_2) \cap \ldots \cap S^i_t(\cdot|r_n). \tag{3}$$

The $\cap$ operator denotes fuzzy intersection, defined by:

$$A(x) \cap B(x) = T(A(x), B(x)),$$

where T is a t-norm as above. There are many options for choosing the T operator, corresponding to different independence assumptions made about the items being combined [10]. In the experiments reported below, we assume that all observations are independent, and therefore we use the product operator for T, which reinforces the effect of consonant observations.

Whatever the choice of T, the belief G_t will be stronger on those locations x that are regarded as possible according both to the prediction, and to all the observations. Eq. (3) thus computes the *consensus* among all the available sources of information about the current robot's location. Notice that the prediction G'_t from the previous belief G_{t-1} is treated as just another source of information in this schema.

[2] Other definitions exist, the one adopted here is taken from [41].

The order in which the observations are combined in Eq. (3) does not affect the result, due to the commutativity and associativity of the T operator. In some cases, however, a normalization may be performed after each individual intersection: it should be noted that in this case the order of combination becomes relevant, since the normalized intersection operator is not associative. In general, if normalization is performed after each intersection, the result will be most strongly influenced by the fuzzy locations that have been combined last. This behavior may be desirable in some domains, e.g., in highly dynamic domains in which more recent observations should be regarded as more reliable. The RoboCup domain used in one of our testbeds is an example of such a domain.

3.5. Overall localization process

We now have all the ingredients needed to define our fuzzy localization process. Below we describe this process in abstract terms; a concrete instantiation of this process will be shown in the next section.

Our fuzzy localization process follows the schema of the well-known predict-observe-update cycle of recursive state estimators [47], where the robot's location is estimated using relative and absolute noisy perceptions, and where uncertainty representation plays a crucial role.

The overall process is graphically shown in Fig. 2. The process follows the classical structure of a fuzzy system, consisting of a fuzzification stage, a fuzzy inference stage, and a defuzzification stage. The fuzzy inference stage, on the other hand, follows the classical structure of a recursive state estimation: a (fuzzy) estimate of the current state is maintained through a predict-update cycle. A dual way to interpret this process is to see it as a classical recursive state estimator, which has been "fuzzified" by embedding it at the core of a classical fuzzy system architecture.

In more detail, the fuzzification part takes as input raw sensor data that provide two types of measurements: exteroceptive and proprioceptive. Exteroceptive measurements come from sensors that observe the status of the environment outside the robot, e.g., vision sensors, and induce information about the absolute position of the robot. Proprioceptive measurements come from sensors that observe the status of the robot's parts, e.g., motion encoders, and induce information about relative displacement. For both types of measurements, the induced items of information can be affected by uncertainty in several ways. These items of information are represented by the fuzzy sets S_t^i for the exceroceptive sensors, and by the fuzzy set (structuring element) B_t for the proprioceptive sensors, as discussed above. The process of building these fuzzy sets from the available sensor data at each time t is called "fuzzification" in the figure, for its resemblance with the corresponding phase in a typical fuzzy system architecture.

The "fuzzy reasoning" phase takes the fuzzy sets produced by the above fuzzification at time t, together with the location belief G_{t-1} computed in the previous cycle, and produces the new location belief G_t in two steps. First, the action model B_t built from proprioceptive data is used, through formula (2) above, to generate a predicted belief G'_t from the previous belief G_{t-1}. Second, the predicted belief is combined, though formula (3), with the fuzzy sets $S_t^1, \ldots, S_t^n$ built from the exceroceptive data, resulting in the updated belief G_t. A key assumption behind this cycle is the Markov assumption, which allows us to only consider the belief state at time $t-1$ in order to compute the belief state at time t, ignoring all the previous states.

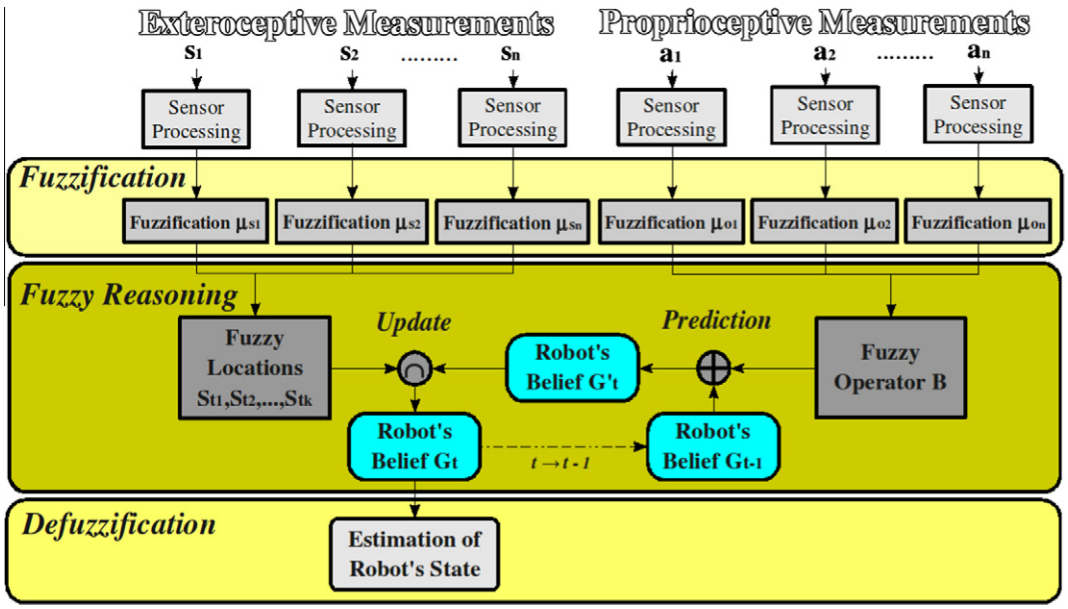


Fig. 2. Fuzzy localization framework. The process follows the typical *Fuzzification–Fuzzy reasoning–Defuzzification* stages of fuzzy systems. The *Fuzzy reasoning* stage follows the typical predict-update cycle of recursive state estimators.

The G_t fuzzy set represents the robot's belief about its own location in the global frame at time t, which is typically needed by the other sub-systems in the robot. Many systems, however, need a point estimate of the robot's location as opposed to a fuzzy one. The purpose of the last phase in our process, "defuzzification", is to extract this estimate, e.g., by computing the center of gravity of the G_t fuzzy set. Note that this computation is only done for the purpose of producing a crisp output, and it does not influence the localization process. In particular, defuzzification does not modify the G_t set that will be used in the next localization cycle.

In addition to the above crisp value, other measures can be extracted from the G_t estimate that give an indication of the quality of this estimate. Two measures that we use in our implementation are:

$$\mathrm{Focus}(G_t) = \frac{\|X\| - \|\mathrm{core}(G_t)\|}{\|X\| - 1} \tag{4}$$

and

$$\mathrm{Rel}(G_t) = 1 - \inf_{x \in X} G_t(x). \tag{5}$$

The first one indicates how much the information in G_t is *focused* on a small area, as opposed to being spread over many possible locations. This is obtained by measuring the size of the core relative to the whole space. In practice, core (G_t) is approximated by a bounding box that contains the cells whose value exceeds a given relative threshold, and $\mathrm{Focus}(G_t)$ is computed so that it is 1 when core (G_t) is a single cell, and it is 0 when core (G_t) is the whole space.

The second measure, $\mathrm{Rel}(G_t)$, indicates how much the information in G_t is *reliable*. Recall from Fig. 1(c) that unreliable location information is represented by introducing a uniform belief over the entire space, representing the fact that the location might be everywhere irrespective of the information. Accordingly, $\mathrm{Rel}(G_t)$ measures the amount of belief which is uniformly allocated to all positions in the space, that is, the minimum value of $G_t(x)$ over the full space X.

We consider that G_t provides an overall good estimate when both the value of $\mathrm{Focus}(G_t)$ and of $\mathrm{Rel}(G_t)$ are high. In the ideal case, when G_t is a distribution with value 1 in a single cell and 0 everywhere else, both values are 1. These measures can be extremely useful to the other sub-systems of the robot. For example, in our RoboCup application, if the robot must reach a strategic position but the reliability measure is too low, the robot stops and tries to acquire visual landmarks in order to improve its self-localization. If, on the other hand, the reliability is high but the focus is low, the robot may not need to improve its estimate if an approximate positioning is sufficient for the task. In a different situation the robot may even act regardless of the quality of self-localization: for instance, when it perceives both the ball and the opponent net in front of it, the robot will kick the ball irrespective of the quality of localization.

4. Fuzzy grid based self-localization

We now describe a particular instance of the general fuzzy self-localization process described in the previous section. In this instance, both the robot's belief and fuzzy locations are represented on a regular grid, and some simplifications are made in order to reduce the computational cost of the method.

4.1. Representing fuzzy locations

Fuzzy locations were defined in the previous section in general terms. When it comes to concrete implementation, one way to represent fuzzy locations is by using a fuzzy position grid. This is an n-dimensional array that represents a tessellation of the robot's workspace X^n: each cell $(i_1,\ldots,i_n)$ in the array represents a region in this tessellation, centered on $(x_1,\ldots,x_n)$. We associate each cell with a number in $[0,1]$, which represents the degree of possibility that the object is located inside the corresponding cell. In the rest of this paper, we focus our attention to regular square tessellations in either 2 or 3 dimensions.

In order to represent the robot's pose, we need a three-dimensional grid, including two dimensions for the (x,y) position and one for the orientation θ. Using a full 3D representation, however, would lead to a relatively high computational complexity, both in time and space. To mitigate this, we use a 2D grid to represent the (x,y) position, and we represent the uncertainty in the robot's orientation by associating to each cell (x,y) a trapezoidal fuzzy set $\mu_{x,y}$ of height $h_{x,y}$ – see Fig. 10 below (top) for an example. The trapezoidal fuzzy set $\mu_{x,y}$ associated to each cell is defined on the $[-\pi,\pi]$ interval, and it is represented by its center θ and by the size of its core and support. The core and support respectively represent the imprecision and the vagueness in the orientation. The height of the trapezoid measures the overall degree of possibility that the robot is located at (x,y) irrespective of its orientation. As a limit case, a trapezoid $\mu_{x,y}$ of height $h = 0$ (i.e., $\mu_{x,y}(\theta) = 0$ for any θ) indicates that the robot cannot be possibly located at (x,y), no matter what its orientation.

For any given cell (x,y),[3] $\mu_{x,y}$ can be seen as a compact representation of a full possibility distribution over the cells $\{(x,y,\theta) \mid \theta \in [-\pi,\pi]\}$ of a full 3D grid. We call this representation a $2\frac{1}{2}$D fuzzy grid. Assuming angular resolution of one degree, a $2\frac{1}{2}$D fuzzy grid reduces the complexity of two orders of magnitude with respect to a 3D one. The price to pay is the inability to handle multiple hypothesis about the robot's orientation at a given (x,y) position. Notice however that multiple hypotheses about the robot's (x,y) position can still be represented.

[3] For notational simplicity, we indicate by (x,y) the cell (i,j) which is associated to the region in space centered on (x,y).

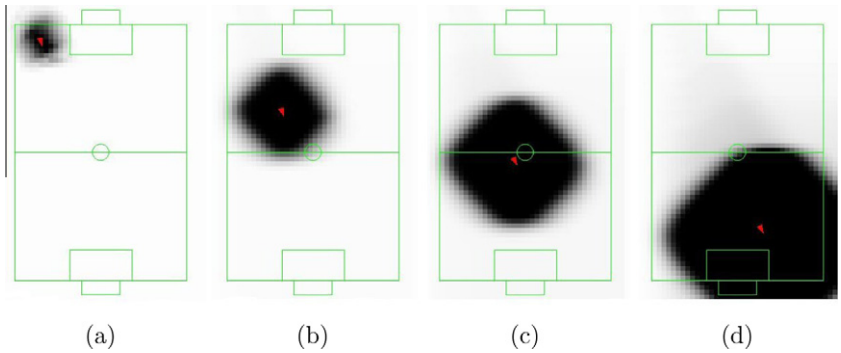


Fig. 3. Example of the *action model* of the fuzzy self-localization method using a grid. The fuzzy robot's belief is translated and rotated (motion model), and blurred (uncertainty model) using tools from fuzzy image processing.

In our implementation, the robot's belief G_t about its own pose is represented by a fuzzy location in the above format over a $2\frac{1}{2}$D possibility grid. Fig. 3 shows four examples of fuzzy locations: the grey level in each cell indicates the height of the trapezoid in the cell, that is, the degree of possibility that the robot is located at that cell's position (darker cells indicate higher degrees). The shape of the orientation trapezoid in each cell is not represented for graphical simplicity. Below, we shall see how the information provided by sensor observations is also represented by fuzzy locations over a $2\frac{1}{2}$D grid, and how all these items of information are processed in the predict-observe-update cycle of our fuzzy recursive estimator.

4.2. Representing the action

When the robot moves, its belief G_{t-1} is updated to G'_t (prediction stage) through Eq. (2) above, using a model B of the robot's motion. In our realization of the framework, B is a fuzzy structuring element whose parameters are dynamically computed from the robot's translation and rotation measured by odometry. Dilation of G_{t-1} by B results in a corresponding translation and rotation of the G_{t-1} distribution, followed by a uniform blurring to account for the uncertainty in the above measurement.

In our implementation, the action model B_t is designed to translate, rotate and blur the robot's belief G_{t-1} represented by the $2\frac{1}{2}$D possibility grid. The translation consists in the convolution of the 2D grid, representing the (x,y) position, with a mask constructed from odometric data. When the displacement is larger than the grid resolution, multiple convolution steps are applied. The rotation consists in shifting the center of the trapezoidal fuzzy sets that represent the orientation of each cell. The uniform blurring adds uncertainty in both the position and the angular information. The former is introduced in the 2D grid by the convolution with a 8-connected circular mask, which introduces uncertainty in all directions, whose size depends on the amount of motion. The latter is introduced in the trapezoidal fuzzy sets by increasing the size of the top and support according to the amount of motion.

Fig. 3 shows an example of how the G_t robot's belief is updated as the robot's move, in the case of an AIBO robot in the RoboCup domain (see Section 5.1 below). Note that extensive blurring is performed in all directions: as we shall see, this is due to the fact that the motion of the AIBO robots is affected by large and highly unpredictable errors.

4.3. Representing the observations

At every cycle t, sensor observations are used to update the predicted robot's belief G'_t and produce the new G_t. Each observation typically consists of a vector r representing the range and bearing to a perceived feature. Knowing the position of the feature in the map, this observation induces a belief in the robot about its own position in the environment, represented by the fuzzy location $S_t(\cdot|r)$.

This fuzzy location induced by an observation may include several types of uncertainty that affect the measure, which depend on the sensor and on the type of feature. For example, an observation from robot camera that detects a uniquely identified object is affected by *imprecision* depending on the pixel size, and by *unreliability* due to the possibility that the object has been wrongly identified. These can be modeled in a way similar to the examples in Fig. 1(a) and (c) above, respectively. More precisely, given the observation $r = (\rho, \theta)$ of a detected object at range ρ and bearing θ, we build two fuzzy trapezoids μ_ρ and μ_θ which represent the uncertainty in range and bearing, respectively. Then, we build $S_t(\cdot|r)$ as follows. For any cell (x,y), the orientation trapezoid stored in $S_t(x,y\,|\,r)$ has center α: the angle at which the object would be seen if the robot were placed at (x,y). The height of this trapezoid is set to the value of $\mu_\rho(d)$, where d is the distance at which the object would be seen if the robot were placed at (x,y). The other parameters of this orientation trapezoid are taken from μ_θ.

Fig. 4(a) shows the belief induced by the visual observation of a uniquely identified landmark (the yellow net) in the RoboCup domain. Given the observed range and bearing to the feature, and the limited resolution of the camera, the robot could be located anywhere in the dark circle (albeit with different orientations, not represented here). The circle has a trapezoidal

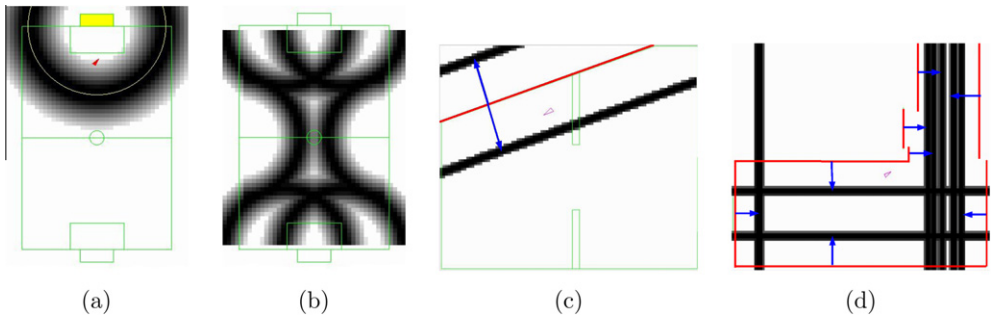


Fig. 4. Examples of belief induced by observations in two different domains: using vision in a RoboCup field, (a) and (b), and using range sensors in a large indoor area, (c) and (d). (a) Unique feature; (b) non-unique feature; (c) unique feature that can be observed from both sides; and (d) non-unique feature that can only be observed from one side.

section. The cells within the top of trapezoid (core) represent those locations which are fully possible given the observation. The cells outside the base of the trapezoid (support) represent those locations which are deemed impossible given the observation. A small uniform "bias" is included in all the cells to account for the possibility that the observation be a false positive, in which case the robot can be just anywhere (unreliability).

As another example, consider the case in which a non-uniquely identified object is observed through the robot camera. The corresponding fuzzy location is affected by *ambiguity*, since multiple locations are possible depending on what is the actual object which has produced the observation. Fig. 4(b) shows the belief induced by the visual observation of a white-line corner on the ground in the RoboCup domain. Since there are several white-line corners in the environment that could be associated to the observed feature (six, in this case), the induced belief is a multi-modal distribution that corresponds to the fuzzy union of six possible fuzzy locations. Fig. 4(c) and (d) shows two additional examples of belief induced by ambiguous observations in a different domain: a robot equipped with a range sensor in a large indoor environment – see Section 5.1 below. In (c), the sensor has detected a uniquely identified wall, but the robot could be on either side of the wall. In (d), the identity of the wall cannot be established.

It is important to note that the above way to represent ambiguous observations does not require a separate data association step. Instead, data association is an integral part of the modeling of an observation, and multiple data association hypotheses are implicitly represented in the corresponding $S_t(\cdot|r)$ fuzzy location, and are then automatically incorporated into the fusion process. Data association is one of the problematic aspects in most current self-localization techniques, and one of the most current reasons for failure.

4.4. Localization process

The top-level grid based localization process is an instance of the general localization process outlined in the previous section. More concretely, the process is summarized as follows.

0 **Initialize:** When the localization system is initialized ($t = 0$), the G_t distribution on the $2\frac{1}{2}$D fuzzy grid is set to represent total ignorance: at every cell (x,y), $\mu_{x,y}$ is assigned the uniformly 1 set – that is, $\mu_{x,y}(\theta) = 1$ for any x, y, θ.
1 **Predict:** At time t, use the data from odometry to evaluate the motion between $t-1$ and t, and compute a fuzzy structuring element B_t that represent this motion, together with the corresponding noise. Compute G'_t from G_{t-1} and B_t according to Eq. (2).
2 **Observe:** For each observation r_i collected between $t-1$ and t, compute a fuzzy distribution $S^i_t(\cdot|r_i)$ on the $2\frac{1}{2}$ grid using the corresponding sensor model.
3 **Update:** Compute the new robot's belief G_t by combining all the above distributions $S^i_t(\cdot|r_i)$ and the G'_t according to Eq. (3). Normalize the resulting distribution.
4 **Defuzzify:** Compute a crisp position estimate g_t of the robot's position by taking the center of gravity of G_t, together with the two quality indicators (Focus and Rel) defined in Eqs. (4) and (5) above.
5 **Loop:** Repeat steps 1–4 above.

After the initialization step (0) the grid represents total ignorance, and it will remain so until an observation is made: predictions from total ignorance will simply result in total ignorance. As soon as the first observation is made, the information in the grid will be updated. This typically results in a more informative distribution, that is, one with lower values.

The update step (3) follows Eq. (3) in general, combining the predicted belief G'_t with the fuzzy distributions $S^i_t(\cdot|r_i)$ from observations. In practice, in our $2\frac{1}{2}$D grid, we use the product operator to intersect the trapezoidal (orientation) fuzzy set in each cell of G'_t with the corresponding one in $S^i_t(\cdot|r_i)$. We use product t-norm because it reinforces the effect of consonant

observations [12]. Since the result of intersecting two trapezoids is not necessarily a trapezoid, we approximate this result by computing the outer trapezoidal envelope of the intersection. This is then stored in the corresponding cell of G_t.

Fig. 5 shows an example of the update step 3 above, in a case where the previous belief G_{t-1} was the empty (uniformly 1) distribution. This corresponds to performing an initial (global) localization with no prior estimate. This example is taken from the RoboCup domain – see Section 5.1 below. The three distributions correspond to the beliefs induced by three distinct visual observations. The first and third distributions come from the observation of two uniquely identified features: the yellow net, and a colored landmark. The second distribution comes from the observation of a non-unique feature: a T-intersection between the white lines painted on the ground. The combination of these three fuzzy locations gives the robot's belief G_t represented on the right.

Fig. 6 shows another example of initial localization, taken from the indoor domain – see Section 5.1. The four distributions correspond to the beliefs induced by four range observations produced by four walls. Each observation is ambiguous since the walls cannot be uniquely identified, and it induces a fuzzy location with a high degree of ambiguity. The intersection of these four fuzzy locations, however, produces a uni-modal and fairly focused fuzzy location, shown at the right. We emphasize once again that no separate data association step is needed in our approach, but association happens implicitly through the intersection operation performed in the update step.

Finally, it should be noted that the above process has nice computational properties, since all the operations involved (computation of the action and sensor models, fuzzy dilation, fuzzy intersection, and center of gravity computation) are linear in the number of cells.

5. Experimental evaluation

We now describe several sets of experiments performed with the aim to validate and evaluate the proposed method. The experiments were performed on very different platforms and environments, in order to test our method under very different conditions. In what follows, we report experiments performed on a small legged robot using vision data, and on a larger wheeled robot using laser range data. These platforms present interesting differences in terms of the types and sources of uncertainty in the sensor data.

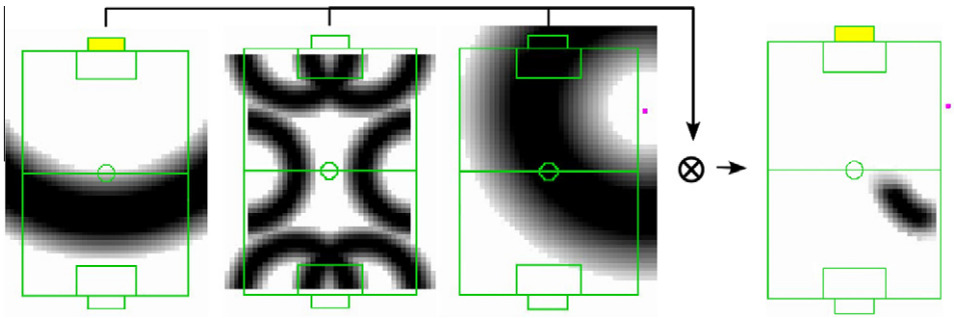


Fig. 5. Example of the update phase: belief resulting from the intersection between the beliefs induced by two unique and one non-unique feature.

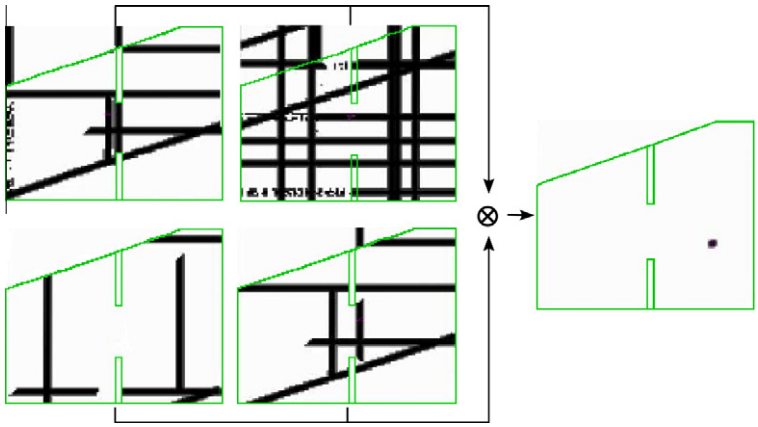


Fig. 6. Another example of the update phase: belief resulting from the intersection between the beliefs induced by four walls (non-unique features).

5.1. Robotic platforms

The legged robot is an *AIBO ERS-7*, a commercial four-legged platform manufactured by Sony; it includes an embedded computer, a low-resolution color camera, and 18 actuated joints with encoders. The main sensor used in our experiments is the camera, used to detect a fixed set of environmental features. The wheeled robot is an *ATRV-Jr*, manufactured by iRobot; it includes a computer, a stereo camera system, a ring of ultrasonic sensors, a SICK LMS laser rangefinder, and a powerful differential drive mechanism with skid steering. The main sensor used in our experiments is the laser rangefinder, which gives a 180 degrees frontal view and provides high accuracy ranges if it is not pointing to a transparent or to a fully reflective surface. We have implemented our fuzzy localization method on several other platforms, including a two-legged humanoid robot *Nao* manufactured by Aldebaran Robotics. Fig. 7 shows some of the used platforms. In this section, we only report the results obtained on the AIBO and the ATRV-Jr robots since these are representative of all the others.

Both platforms provide sensor data which are affected by uncertainty to a large extent, although for different reasons. In the case of the AIBO robot, there is a large uncertainty in the odometric data. AIBO odometry is estimated in open-loop based on a simple motion model. However, due to the inherent problems of legged locomotion, this estimate may be far away from the real motion performed. Sources of error include the unpredictable slippage of the feet, the amplified effect of even minor disturbances in the timing of each leg motion, the difficulty to control the body posture which may cause stumbles and dragging, the non-modeled dependency of the walking performance on the type of surface, and the influence of the battery level. Notice that these errors affect both the measurements of rotation and of translation. In addition to odometric uncertainty, visual landmark observations are also affected by large uncertainty. This is mainly due to the low quality of the camera, which is used as sole exteroceptive sensor, and to noise and errors in estimating the camera position based on the encoders in the robot's neck joints. These factors strongly affect the estimate of the distance to observed features, and to a smaller extent the estimate of their bearing.

In the case of the ATRV-Jr robot, the main source of odometric uncertainty is the use of skid steering: two wheels on each side are linked, which inevitably results in sliding whenever a rotation is performed, making odometric estimation quite unreliable in these cases. Interesting, this type of motion uncertainty is different from the one that affects the AIBO – e.g., odometric estimation during straight motion is quite reliable on the ATRV-Jr, while it is not on the AIBO.

5.2. Environments

The experiments were also performed in very different environments. The legged robot operates in the standard environment used in RoboCup for the Four-Legged League in 2006 [48], and its tasks involve navigation among other robots and ball manipulation. The features of interest in this environment are the standard color-coded objects used in this league (nets, landmarks, ball, and other robots), as well as the white lines painted on the field ground. Both types of features are detected using the on-board color camera.

The vision techniques that we used in the RoboCup domain are divided into two groups; (i) processing based on colors and (ii) processing based on features [13,49]. The first group aims to detect and identify the color-coded objects in the environment, using prior information about their dimension, location and colors. The second group of techniques aims to detect and identify field features in the environment. In all cases, the vision processes have been developed to run in real time using the limited processing on-board resources available in the AIBO robots.

The environment used for the wheeled robot is a relatively large indoor environment, and the only task considered here is navigation. The features of interest are the environment walls, detected by the extraction of line-segments from the readings of the laser rangefinder. We use a *Split-and-Merge* algorithm for this, since its high speed makes it adequate for real-time applications [50]. More precisely, we use a simplified version of this method, namely *Iterative-End-Point-Fit* method [51]. The local coordinates of the wheeled robot and the line-segments sensed in an indoor environment are shown in Fig. 8(b), where dashed lines represent the sensed walls and the arrows the parameters of the line segments in normal form.

Fig. 7. Some of the platforms on which our method has been tested: (a) AIBO; (b) ATRV-Jr; and (c) Nao.

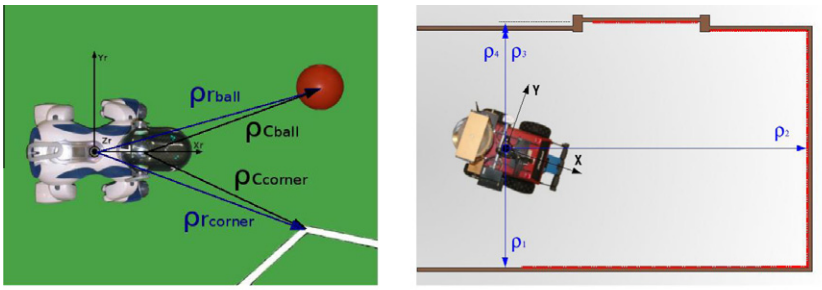


Fig. 8. Example of the perceptions given by the (a) vision and (b) range systems of the legged and the wheeled robot, respectively.

The readings from the laser are affected by noise that should be correctly represented into the line-segments, so that it can be taken into account in the localization algorithm. To do so, we follow the proposal by Gasós [22] who represents the uncertainty in the location of the segments using *fuzzy segments*. An interesting feature of fuzzy segments is that a degree of similarity can be easily defined, thus allowing to match nearby segments in an effective way. A fuzzy segment represents the uncertainty of the ρ parameter of normal form of the line segment. This uncertainty includes the effects of the noise in the reading as well as in the uncertainty in the robot's pose when the readings have been collected.

The fuzzy models for both platforms have been designed from heuristic knowledge based on authors' experience, who have estimated the approximate type and extent of the uncertainty affecting perception and odometry. The *fuzzy sensor models* consist in two trapezoidal fuzzy sets, μ_ρ and μ_θ, representing the uncertainty in range ρ and orientation θ to the perceived object. These fuzzy sets are defined by $\langle \alpha, \Delta, \delta, h, b \rangle$, where α is the center of the set, Δ and δ are the width of the core and support respectively, h is the height and b is the bias (representing unreliability). In the case of the AIBO, the values of Δ and δ for μ_ρ are proportional to the distance to perceived object; this is because distance is estimated based on the object's elevation in the image, a method which incurs in larger errors as the distance increases. The values of Δ and δ for μ_θ depend on the size of perceived objects and on the resolution of the camera, since errors in the estimation of orientation depend mostly on pixelation. The height h is 1 for all sensor models, while the bias value b depends on the degree of reliability provided by the image processing routines. In the case of the ATRV-Jr, the parameters of the μ_ρ and μ_θ fuzzy sets are provided by the fuzzy segment extraction routines [22], which are rather accurate thanks to the precision of laser sensor. As for the *fuzzy action models*, these depend on two coefficients, b_ρ and b_θ, which weight the amount of blurring depending on the rotation and translation received. The AIBO platform always introduces high uncertainty when moving, while the ATRV-Jr is rather precise in linear motion but introduces high motion error when it rotates.

The above parameters (μ_ρ, μ_θ, b_ρ and b_θ) are all the parameters that influence the performance of our method. In our experience, the setting of these parameters is not critical: the experiments reported below have been made using rough settings based on intuitive knowledge, and slightly different settings did not results in substantially different performance. A more systematic analysis aimed at determining the robustness of the results with respect to the values of the parameters might be part of our future work, but it was not deemed necessary at this stage.

Ground truth measurements were obtained during all the experiments in order to enable the quantitative evaluation of the quality of the estimates computed by the tested methods. In particular, for the experiments with the wheeled robot we have used an external vision system, described below; while for the experiments using the wheeled robot we have used triangulation from manual measurements.

6. Results on a legged robot

We now present the results of the experimental evaluation performed on the AIBO legged robots. In the experiments, the robot used the method proposed in this paper to self-localize based on the observable features available in a standard RoboCup field. These features include artificial color-coded landmarks (beacons and goals) and natural field features like the white lines on the ground and their intersections. The purpose of the experiments was to test the capabilities of our localization method under several conditions, including both normal navigation and kidnapping. Kidnapping is not unusual in this RoboCup league: when robots are penalized, they are manually moved by the referees outside the field for some time, and then manually moved back into the field in some unpredictable position.

Two experiments are reported below: (i) a typical navigation sequence of a goalkeeper that comes off its area, and (ii) a kidnapping situation. In both experiments, we evaluate the results quantitatively with the help of an external vision-based tracking system to establish the ground truth. We also run both experiments using a state-of-the-art stochastic localization method, a variant of Monte Carlo localization (MCL), and compare the quality of the estimate produced by that method with the one produced by our fuzzy grid based approach.

6.1. Experimental set-up

The set-up is the official field used in the *Four-Legged Robot League* from RoboCup 2006. In order to evaluate the quality of the localization during the experiments, an external vision system has been developed to provide us with ground truth. This system can track the position and orientation of several color-coded markers placed on the back of the robots. The system is composed of an overhead camera with wide angle lens, mounted on an aluminum structure at a height of 2.5 m – see Fig. 9. The wide angle lens allows the camera to cover half of the field from its mounting point, but introduce distortions in the image which must be compensated for a correct tracking of the markers. In addition, other sources of error should be considered, including pixelation, errors in color segmentation, and errors in the transformation from the camera plane to the world frame. To estimate these errors, we have collected measurements from markers placed on a known 8×10 grid on the field. We have performed more than fifty experiments using different camera and color segmentation calibrations, and we have determined that the position error from the external vision system is less than 5 cm in average, and bounded to 10 cm. The heading error is less than 10° in average, and bounded to 18°. We consider that these errors are small enough to justify the use of this external vision system as the ground truth in this domain.

6.2. The reference method

The experiments reported in this section have a double aim: to show that the proposed fuzzy localization method is able to effectively address the self-localization problem, and to compare this method with a probabilistic localization method.

The reference method is a variant of Monte Carlo localization (MCL) [29], in which the probability density is represented by maintaining a set of samples that are randomly drawn from it. This variant has been developed by the Upennalizer team for the same RoboCup domain as the one considered here [52], and it uses a hybrid representation of the probability density to reduce computational cost. The pose probability is factorized as a distribution over a discrete set of angles and continuous translational coordinates; the distribution over poses (x,y,θ) is first generically decomposed into the product $P(x,y,\theta) = P(\theta) \cdot P(x,y|\theta) = \sum_i P(\theta_i) \cdot P(x,y|\theta_i)$, which is a kind of Rao-Blackwellization of the state space [53]. The distribution $P(\theta)$ is modeled as a discrete set of weighted samples θ_i, and the conditional likelihood $P(x,y|\theta)$ as simple two-dimensional Gaussian. This approach has the advantage of combining discrete Markov updates for the orientation with Kalman filter updates for the translational degrees of freedom.

In order to make the comparison between our fuzzy localization technique and the probabilistic reference method as fair as possible, we have used similar sensor and action models. How to do this, however, is not obvious since fuzzy and probabilistic techniques are semantically different: we interpret fuzzy sets to represent degrees of possibilities, while probabilities are more naturally interpreted in terms of stochastic events. Moreover, stochastic methods need sensor models based on frequencies, hence the probability function that models the sensor should be experimentally obtained, whereas methods based on fuzzy logic make use of qualitative sensor models. In our experiments, we have ignored these semantic differences, and have used probabilistic sensor models that directly reflect the fuzzy ones. Fig. 10 shows an example of the sensor models used and the relation between them. The fuzzy sensor model incorporates a combination of several types of uncertainty, including: imprecision, represented by the width of the core of the fuzzy set; vagueness, represented by the slopes of the fuzzy set; and unreliability, represented by the uniform "bias" in the fuzzy set. The height h indicates the degree of possibility that the robot is located within the width of the core. This fuzzy sensor model can be read as "the object is believed to be approximately within the core, but this belief might be wrong". The stochastic sensor model is represented by a two-dimensional Gaussian function, whose parameters are chosen so that the core and the support of the fuzzy model correspond to two and four standard deviations of the stochastic Gaussian function, respectively. In all our experiments the height h was set to 1, although it can be set to a lower value in other sensor models (as it is the case in the example shown Fig. 10). Notice that there is no relation between the value of h and the height of the Gaussian Probability Density Function (PDF). In particular, in the fuzzy case there is no constraint that the values integrate to one. Also notice that the fuzzy sensor models used here are symmetric in order to reflect the Gaussian PDF. However, symmetry is not a necessary constraint, and asymmetric models can be used without any change in our method.

Fig. 9. Experimental set-up of the experiments using the legged robot: (a) aluminum frame holding the external vision system, (b) overhead camera with wide angle lens, and (c) a robot wearing a color-coded mark.

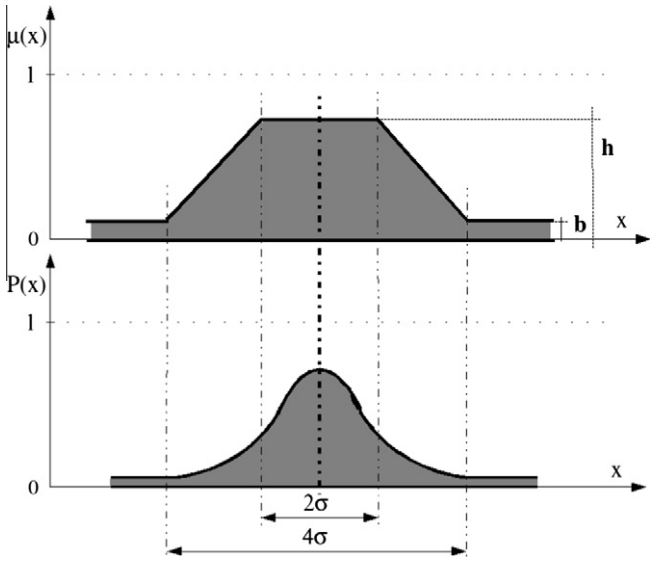


Fig. 10. Sensor models used to perform the comparison experiments; correspondence between (upper) fuzzy set and (lower) stochastic Gaussian distribution.

In the rest of this section, we report the experiments performed to evaluate our fuzzy localization method, the comparison with the reference method, and a discussion of the results.

6.3. Experiment 1: goalkeeper coming off its line

This experiment reproduces a common situation in RoboCup games: the goalkeeper comes off its area to narrow the angle of an opponent player, and then needs to returns to its place. The experiment was performed by letting the robot execute the

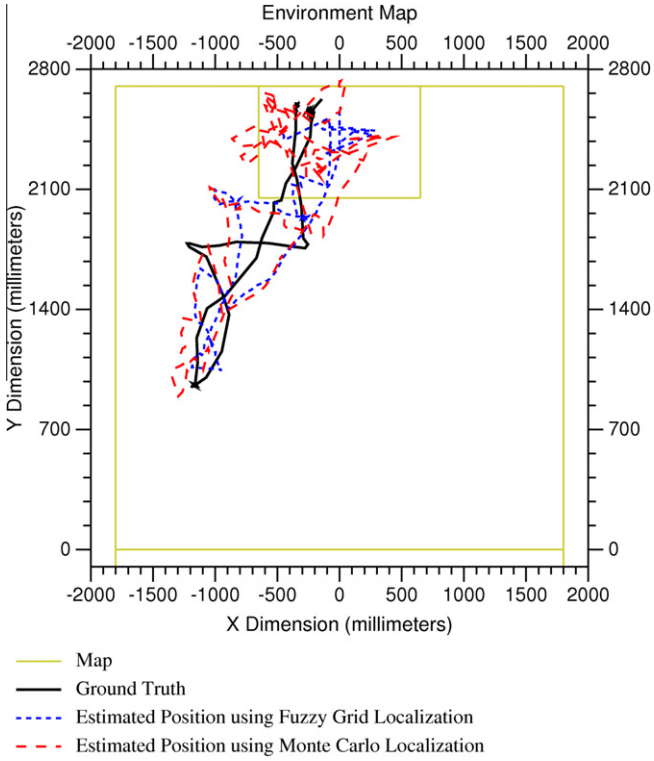


Fig. 11. Goal-keeper experiment; ground truth (dark continuous line) and estimate trajectories using fuzzy grid based (blue dotted line) and Monte Carlo (red dashed line) localization methods. (For interpretation of the references in colour in this figure legend, the reader is referred to the web version of this article.)

goal-keeper code used by our team during actual competitions. The robot is initially placed in the goal area, facing the opposite net, then it is forced into the "cut-opponent" behavior, which makes it come off its line to narrow down the opponent's angle, and it is finally forced into the "go-back" behavior, which makes it return to its goal area. The experiment is initialized with a belief uniformly distributed on the whole field, i.e., the robot does not know its own initial location. Then, the robot starts scanning its surroundings by moving its head from left to right. As soon as a feature is sensed, it is incorporated into the localization process. Fig. 11 shows the estimates of the robot's pose during the experiment using our fuzzy localization method and using the reference method, together with the ground truth provided by the external vision system.

The quantified errors of the two methods are shown in Fig. 13 below. The magnitude of the position error is smaller in average using the fuzzy method than using the reference method. No noticeable difference can be observed for the heading errors.

6.4. Experiment 2: kidnapping

The next experiment is intended to verify the ability of our method to cope with "kidnapping" situations, i.e., the robot is manually transported from one location in the field to another one, without receiving any sensor data during the displacement. The robot should be able to detect that it has been kidnapped, and to find out its new location. In this experiment, the robot does not perform any walking: only measurements from the camera are used, which are collected while the head scans the environment horizontally. As in the previous experiment, the localization methods are initialized with a belief uniformly distributed along the whole field, i.e., the robot does not know its own initial location.

The experiment is performed as follows. First, the robot is placed at the initial position, and it scans the environment for some seconds thus incorporating enough observations into the localization process to assess its initial position; then the robot is kidnapped and transferred to another, pre-defined location. Each experiment is repeated three times. Fig. 12 shows the locations used for this experiment, together with the features that can be observed by the robot from these locations. The locations have been chosen so that the robot can see different types of features at different locations, given its limited field of view: Table 1 lists what features are observable from what location.

Fig. 12. Kidnapping sequence, including robot's locations and sensed objects.

Table 1
Robot's positions in kidnapping experiment (millimeters and degrees), and the sensed features from such positions.

	Initial position	Kidnapping 1	Kidnapping 2	Kidnapping 3
Kidnapping experiment sequence				
(x,y)	(−282,2489)	(−478,778)	(1010,524)	(−1068,356)
θ	−122	146	113	−52
NET 1	X			X
NET 2		X	X	
LM 1	X			X
LM 2			X	
LM 3	X	X	X	X
LM 4	X	X	X	
Ci	X			
T3				X

The position and heading errors of the experiment are shown in Fig. 13. Like in the previous experiment, errors are computed with respect to the ground truth provided by the external vision system. The vertical dashed lines indicate the moments when the robot was kidnapped. These moments are also clearly visible in the peaks in the position error caused by the sudden discrepancy between the real and the estimated location. After the robot is kidnapped, it continues to collect measurements and to incorporate these into the localization process. The new measurements are clearly inconsistent with the previous robot's belief, and eventually both the localization methods manage to correct the robot's belief and reduce both the position and the heading errors.

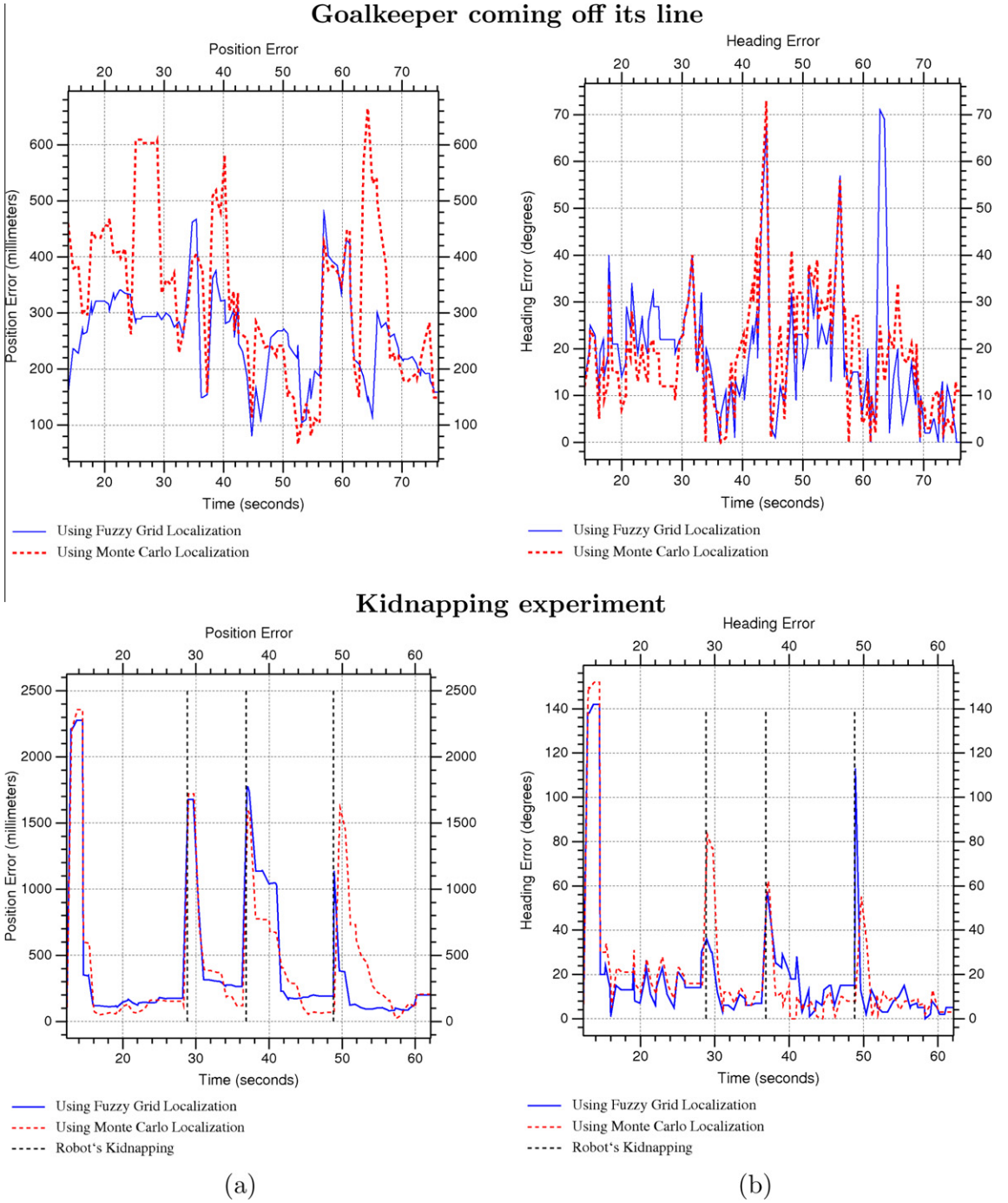


Fig. 13. Comparison between fuzzy grid based and a variant of Monte Carlo localization methods in different experiments.

The data show that the fuzzy localization method is able to reduce the position error after the kidnapping faster than the reference method, but that the reference method eventually provides a more accurate estimate. This is mainly due to the fact that the reference method works on a continuous metric representation, while the fuzzy localization method is based on a grid with coarse resolution, which clearly limits its accuracy. For what concerns the heading error, no major difference could be observed between the two methods. After the second kidnapping, the fuzzy localization method takes longer than the reference method to converge again to the correct orientation estimate. We attribute this fact to the fact that the orientation between the positions before and after the kidnapping are fairly similar, so new observation lead only to smooth changes in the fuzzy grid. By contrast, when the difference is large the inclusion of new observations into the fuzzy grid leads to an inconsistent situation, represented by a nearly uniform fuzzy estimate G_t; the following observations then have a strong immediate impact on this uniform estimate, causing G_t to converge quickly.

7. Results on a wheeled robot

We now present the results of the experimental evaluation performed on a wheeled platform, the ATRV-Jr robot. We use the fuzzy grid based self-localization method based on features extracted from laser data. In particular, we use *fuzzy segments*, linear features obtained from the observation of wall segments, which include an estimate of their position uncertainty. The purpose of these experiments is to test our localization method on a different type of platform, and on larger environments. Moreover, these experiments show how our method handles perceptual ambiguity, since the same observation of a wall segments can originate from several different walls in the environment.

7.1. Experimental set-up

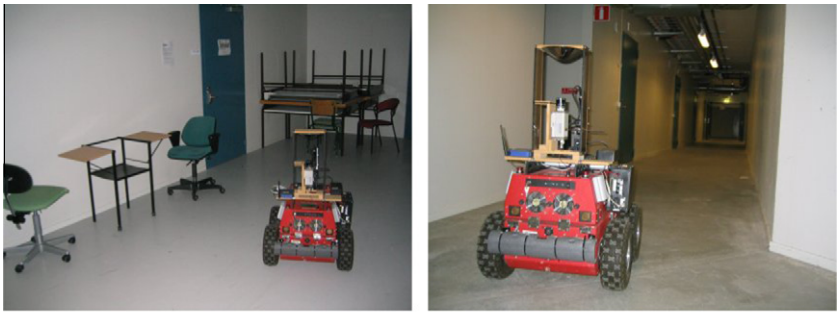
The experiment using the wheeled robot was performed in a relative large indoor environment, the basement of a building of approximately 15×43 meters, partially shown in Fig. 14. The environment included long corridors, radiators, doors, and office furniture, and was not modified to perform the experiment. The robot was driven manually using a joystick, with the operator walking behind the robot. The environment was static, i.e., nothing was changed during the experiment except the position of the robot. While the robot moved, the readings from its laser rangefinder were used to generate *fuzzy segments* representing the robot's observation through its fuzzy sensor model, and the readings from its odometric encoders were used to generate the fuzzy structuring element representing the fuzzy action model.

7.2. Indoor experiment

The environment was represented using a fuzzy grid with cell resolution of 10 cm, which was empirically determined as a good trade-off between accuracy and computational cost. In our environment, this resulted in a fuzzy grid of 153×426 cells.

In order to quantitatively evaluate the estimates produced by our method, we have computed a ground-truth reference as follows. We manually inspect the recorded laser data at selected moments during the run, and extract from them the perpendicular distance to the walls. We then determine by visual inspection what laser data correspond to what wall at that moment, and use this information to compute the real position and heading of the robot by manual triangulation. This computation provides us with ground truth measurements at a set of pre-defined strategic locations. The quality of the robot's estimate is evaluated with respect to these measurements.

As in all the previous experiments, our localization method is initialized with a uniform belief distributed along the whole environment: the robot does not know its location at start. Then, the robot starts moving and collecting range measurements. As soon as a feature is sensed or an action is performed, it is incorporated into the localization process. The resulting beliefs and the features sensed are shown in the sequence of the Fig. 15, together with the map provided to the robot, which



(a) (b)

Fig. 14. Wheeled robot in the indoor environment where the experiment is performed.

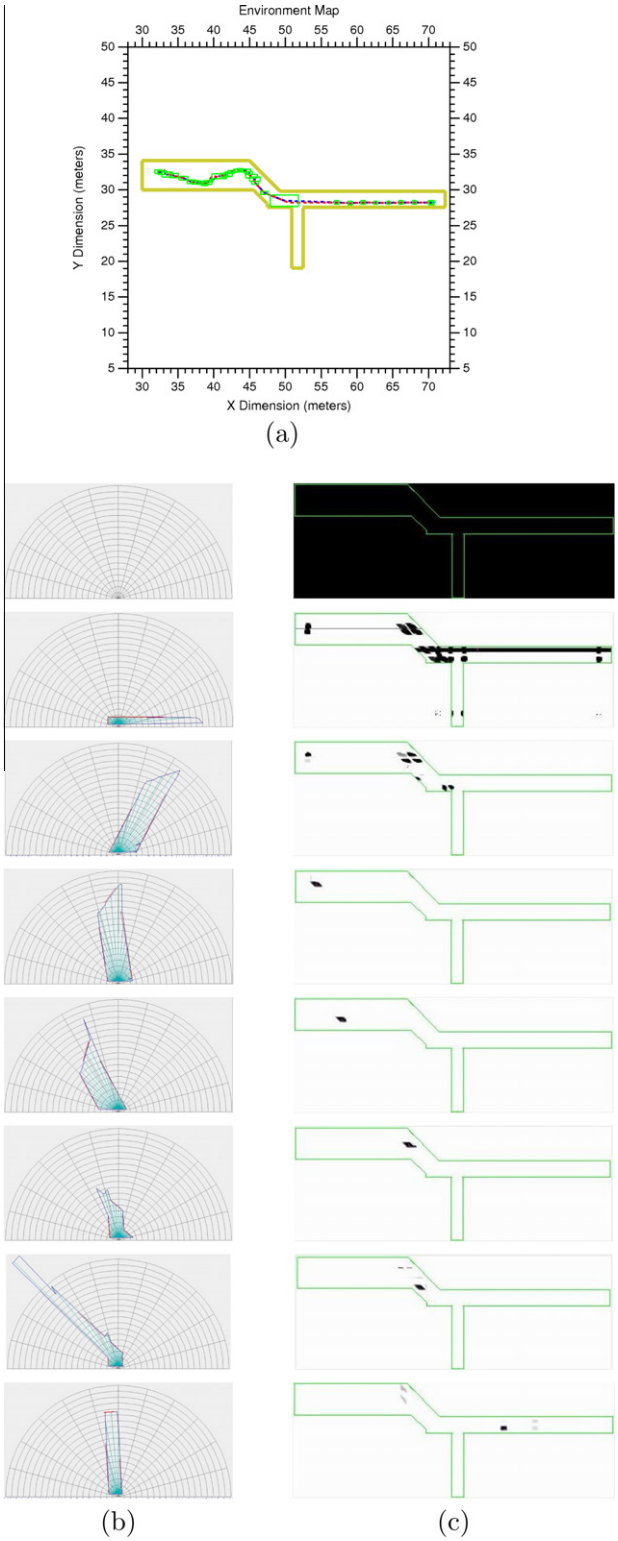


Fig. 15. Experiment of wheeled robot in an indoor environment; (a) ground truth (red dotted line), estimated trajectory (blue dotted line) and bounding boxes (green empty boxes) indicating the estimation of errors at different times using the fuzzy grid based localization method; sequence of sensed (b) fuzzy segments; and (c) estimated robot's belief. (For interpretation of the references in colour in this figure legend, the reader is referred to the web version of this article.)

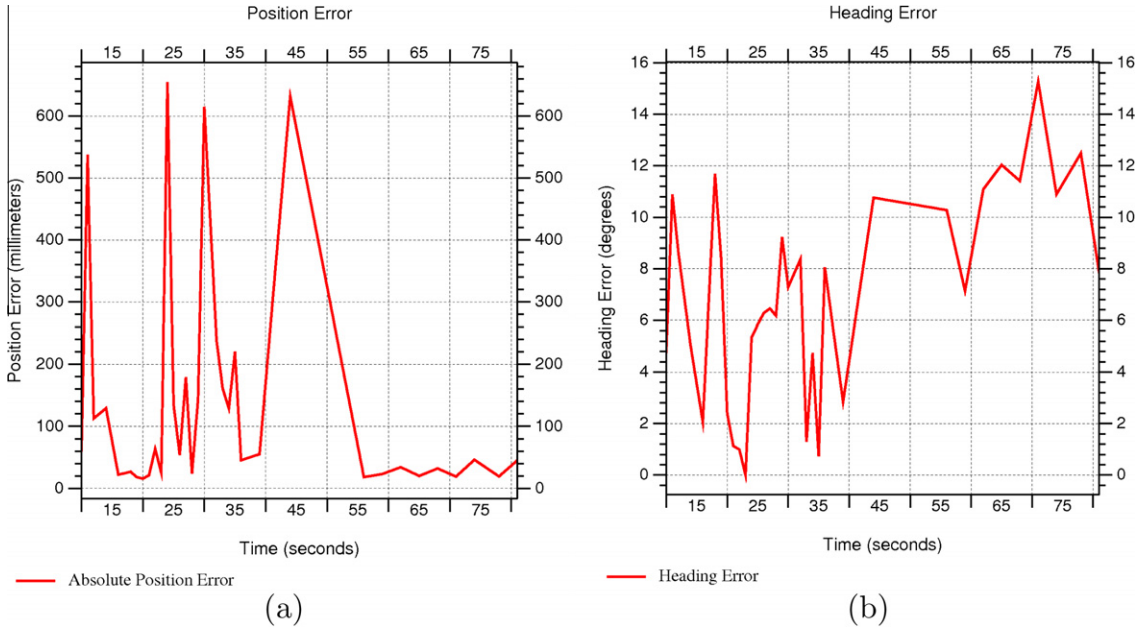


Fig. 16. (a) Position and (b) heading errors of wheeled robot indoor localization using the fuzzy grid based method.

consists of a set of line-segments representing the walls in the environment. The figure also shows the actual and estimated trajectories during the experiment, and the bounding boxes representing the uncertainty of the robot's belief in each dimension during the navigation.

As it can be seen in the figure, the robot needs to collect observations from several different locations in order to obtain an estimate that represents a unique position, since each observation may have multiple possible origins. In fact, the robot cannot arrive to a unique estimate using only the observations taken from the initial position, no matter how many of these. At the start of the sequence shown in Fig. 15(c), the robot's belief G_t is distributed along the whole field; after some observations, G_t presents more than eight possible clouds where the robot could be located, i.e., the localization method is not able to find out the robot's pose up to several seconds navigating. Only after the robot has traveled some distance, the ambiguity is solved and the possible clouds in G_t are reduced to one. In the remaining part of the experiment, the localization method is able to correctly maintain the robot's belief about its location. The actual robot's trajectory, plot in Fig. 15(a), is inside the uncertainty bounding box for the entire duration of the experiment.

Fig. 16 shows the position and orientation errors, computed when the self-localization method provides high reliability value for its position estimate, i.e., once it has converged. The position error shows many peaks. This is due to the fuzzy motion model that we have used, which includes a high degree of uncertainty in the robot's belief in all directions; we have decided to use such a rough motion model to account for the unreliable odometry of our robot, mostly due to the use of skid steering. The robot's belief increases continuously while the robot is in motion, and until a new relevant observation is made. This is a typical problem for most localization methods in long corridors: motion makes the robot's uncertainty to increase in all directions, but observations of the side walls only reduce uncertainty in the lateral direction. Longitudinal uncertainty then grows continuously, until a distinctive feature is observed in the longitudinal direction, e.g., a door, or the end of the corridor. Despite this problem, the results show that the position error is bounded to less than 25 cm for most of the experiment and never larger than 65 cm during the entire experiment. The heading error is less than 7 degrees in average, and it is never larger than 16 degrees.

8. Conclusions

We have presented a localization method for mobile robots based on the use of fuzzy logic. The key idea of our method is to include the predict-update cycle of recursive state estimators as the core of a typical fuzzy system. The key steps to do so are: (1) the robot's location belief is represented as a fuzzy set on a (discretized) representation of the robot's environment; (2) a fuzzification step is used to convert both relative and absolute position measurements to fuzzy sets on the above representation; (3) the predict-update cycle is re-defined in terms of manipulation of fuzzy sets; and (4) a defuzzification step is used to convert fuzzy belief to position estimates to be used by the other sub-systems in the robot. We have described a concrete implementation of our method based on a fuzzy position grid, including several simplifications aimed at reducing its computational complexity. Our implementation runs on-board the resource-limited AIBO robots at 10 Hz.

The main advantages of the presented method are: (i) the use of fuzzy logic allows us to represent different facets of the uncertainty that affects the measurements; (ii) the sensor models are approximate and can be defined from heuristic knowledge; and (iii) ambiguity in observations is handled as part of the method, and does not require a separate data association step. We have presented experiments conducted using different robotic platforms and in different environments. The experiments confirm that our method is able to solve both position tracking and global localization while correctly handling ambiguities. The experiments have also shown that our method provides more stable estimates than a state-of-the-art probabilistic localization method in situations of high motion uncertainty. The price to pay for these advantages is a consequence of its grid based representation: on the one hand, accuracy depends on the size of the cells; on the other hand, a finer grid resolution leads to increased computational cost. In general, the computational complexity of the method grows linearly with the size of the environment, somehow limiting its scalability over very large environments. In the domains considered in this paper, characterized by high uncertainty, these disadvantages are largely compensated by the stability of the method and its robustness.

Our future work includes mainly three points. First, the development of hierarchical representations to improve the scalability of our method to very large environments. Second, a deeper study of suitable confidence measures and an experimental validation of their value as indicators of the real quality, at any moment, of the produced location estimate. Finally, we would like to verify the applicability of our method to an even wider set of robotic platforms, especially in outdoor domains.

Acknowledgments

This work has been supported by DPI2004-07993-C03-02 and DPI-2007-66556-C03-02 CICYT projects, from the Spanish Ministry of Science and Innovation, by the Swedish graduate school in computer science (CUGS), and by the Swedish Knowledge Foundation. We would like to thank the anonymous reviewers for insightful comments and suggestions. We are indebted to Pär Buschka and Zbignew Wasik for their invaluable help with the AIBO implementation.

References

[1] J. Leonard, H. Durrant-Whyte, Mobile robot localization by tracking geometric beacons, IEEE Transactions on Robotics and Automation 7 (3) (1991) 376–382.
[2] L.A. Zadeh, Fuzzy sets, Information and Control 8 (3) (1965) 338–353.
[3] A. Saffiotti, The uses of fuzzy logic in autonomous robot navigation, Soft Computing 1 (4) (1997) 180–197.
[4] K. Tanaka, M. Sugeno, Stability analysis and design of fuzzy control systems, Fuzzy Sets and Systems 45 (2) (1992) 135–156.
[5] K. Tanaka, M. Sano, Trajectory stabilization of a model car via fuzzy control, Fuzzy Sets and Systems 70 (2–3) (1995) 155–170.
[6] R.R. Yager, On a hierarchical structure for fuzzy modeling and control, IEEE Transactions on Systems, Man, and Cybernetics 23 (4) (1993) 1189–1197.
[7] A. Bonarini, M. Matteucci, M. Restelli, Concepts and fuzzy models for behavior-based robotics, International Journal of Approximate Reasoning 41 (2) (2006) 110–127.
[8] H. Liu, G.M. Coghill, D.J. Brown, Qualitative kinematics of planar robots: intelligent connection, International Journal of Approximate Reasoning 46 (3) (2007) 525–541.
[9] H. Liu, G.M. Coghill, D.P. Barnes, Fuzzy qualitative trigonometry, International Journal of Approximate Reasoning 51 (1) (2009) 71–88.
[10] I. Bloch, Information combination operator for data fusion: a compartive review with classification, IEEE Transactions on Systems, Man, and Cybernetics, Part A 26 (1) (1996) 52–67.
[11] S. Benferha, C. Sossai, Reasoning with multiple-source information in a possibilistic logic framework, Information Fusion 7 (1) (2006) 80–96.
[12] P. Buschka, A. Saffiotti, Z. Wasik, Fuzzy landmark-based localization for a legged robot, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'00), Takamatsu, Japan, 2000, pp. 1205–1210.
[13] D. Herrero-Pérez, H. Martínez-Barberá, A. Saffiotti, Fuzzy self-localization using natural features in the four-legged league, in: Robot Soccer World Cup VIII, Lecture Notes in Artificial Intelligence, Lisbon, Portugal, vol. 3276, 2005, pp. 110–121.
[14] D. Fox, W. Burgard, S. Thrun, Markov localization for mobile robots in dynamic environments, Journal of Artificial Intelligence Research 11 (1999) 391–427.
[15] D. Fox, J. Hightower, L. Liao, D. Schulz, G. Borriello, Bayesian filtering for location estimation, IEEE Pervasive Computing Magazine 2 (3) (2003) 24–33.
[16] H. Baltzakis, P. Trahanias, A hybrid framework for mobile robot localization: formulation using switching state-space models, Autonomous Robots 15 (2) (2003) 169–191.
[17] J. Gutmann, W. Burgard, D. Fox, K. Konolige, An experimental comparison of localization methods, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'98), Victoria, Canada, 1998, pp. 736–743.
[18] J. Gutmann, D. Fox. An experimental comparison of localization methods continued, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'02), Lausanne, Switzerland, 2002, pp. 454–459.
[19] J. Borenstein, H. Everett, L. Feng, D. Wehe, Mobile robot positioning: sensors and techniques, Journal of Robotic Systems 14 (4) (1997) 231–249.
[20] A. Saffiotti, K. Konolige, E.H. Ruspini, A multivalued-logic approach to integrating planning and control, Artificial Intelligence 76 (1–2) (1995) 481–526.
[21] G. Oriolo, G. Ulivi, M. Venditelli, Real-time map building and navigation for autonomous mobile robots in unknown environments, IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics 3 (28) (1998) 316–333.
[22] J. Gasós, A. Martín, A fuzzy approach to build sonar maps for mobile robots, Computers in Industry 32 (2) (1996) 151–167.
[23] J. Gasós, A. Martín, Mobile robot localization using fuzzy maps, in: Fuzzy Logic in Artificial Intelligence Towards Intelligent Systems, Lecture Notes in Artificial Intelligence, vol. 1188, 1997, pp. 207–224.
[24] K. Demirli, M. Molhim, Fuzzy dynamic localization for mobile robots, Fuzzy Sets and Systems 144 (2) (2004) 251–283.
[25] A. Saffiotti, L. Wesley, Perception-based self-localization using fuzzy locations, in: Reasoning with Uncertainty in Robotics, Lecture Notes in Artificial Intelligence, vol. 1093, Amsterdam, The Netherlands, 1996, pp. 368–385.
[26] W. Burgard, D. Fox, D. Henning, T. Schmidt, Estimating the absolute position of a mobile robot using position probability grids, in: 13th National Conference on Artificial Intelligence, Menlo Park, AAAI, AAAI Press/MIT Press, 1996, pp. 896–901.
[27] L. Jetto, S. Longhi, G. Venturini, Development and experimental validation of an adaptive extended Kalman filter for the localization of mobile robots, IEEE Transactions on Robotics and Automation 15 (2) (1999) 219–229.
[28] H.H. Lin, C.C. Tsai, Improved global localization of an indoor mobile robot via fuzzy extended information filtering, Robotica 26 (2) (2008) 241–254.
[29] S. Thrun, D. Fox, W. Burgard, F. Dellaert, Robust Monte Carlo localization for mobile robots, Artificial intelligence 128 (1–2) (2001) 99–141.

[30] H. Kose, H.L. Akin, A fuzzy touch to R-MCL localization algorithm, in: Robot Soccer World Cup IX, Lecture Notes in Artificial Intelligence, vol. 4020, Osaka, Japan, 2006, pp. 420–427.
[31] H. Kose, B. Celik, H.L. Akin, Comparison of localization methods for a robot soccer team, International Journal of Advanced Robotic Systems 3 (4) (2006) 295–302.
[32] H. Kose, H.L. Akin, The reverse Monte Carlo localization algorithm, Robotics and Autonomous Systems 55 (6) (2007) 480–489.
[33] J.P. Canovas, K. LeBlanc, A. Saffiotti, Robust multi-robot object localization using fuzzy logic, in: Robot Soccer World Cup VIII, Lecture Notes in Artificial Intelligence, vol. 3276, Lisbon, Portugal, 2005, pp. 247–261.
[34] K. LeBlanc, A. Saffiotti, Multirobot object localization: a fuzzy fusion approach, IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics 39 (5) (2009) 1259–1276.
[35] N.E. Özkucur, B. Kurt, H.L. Akin, A collaborative multi-robot localization method without robot identification, in: Robot Soccer World Cup XII, Lecture Notes in Artificial Intelligence, vol. 5399, Suzhou, China, 2008, pp. 189–199.
[36] J. Santos, P. Lima, Multi-robot cooperative object localization – decentralized Bayesian approach, in: Robot Soccer World Cup XIII, Lecture Notes in Artificial Intelligence, vol. 5949, Graz, Austria, 2010, pp. 332–343.
[37] J. Gasós, A. Saffiotti, Integrating fuzzy geometric maps and topological maps for robot navigation, in: 3rd International Conference ICSC Symposium on Soft Computing (SOCO'99), Genova, Italy, 1999, pp. 754–760.
[38] J. Gasós, A. Rosetti, Uncertainty representation for mobile robots: perception, modeling and navigation in unknown environments, Fuzzy Sets and Systems 107 (1) (1999) 1–24.
[39] M. Begum, G.K.I. Manna, R.G. Gosinea, Integrated fuzzy logic and genetic algorithmic approach for simultaneous localization and mapping of mobile robots, Applied Soft Computing 8 (1) (2008) 150–165.
[40] L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets and Systems 100 (1) (1999) 9–34.
[41] I. Bloch, H. Maître, Fuzzy mathematical morphologies: a comparative study, Pattern Recognition 28 (9) (1995) 1341–1387.
[42] G. Matheron, Eléments pour une théorie des milieux poreux, Masson, Paris, 1967.
[43] J. Serra, Image Analysis and Mathematical Morphology, Academic Press, London, 1982.
[44] I. Bloch, A. Saffiotti, Why robots should use fuzzy mathematical morphology, in: 1st International ICSC-NAISO Congress on Neuro-Fuzzy Technologies, La Havana, Cuba, 2002.
[45] S. Weber, A general concept of fuzzy connectives, negations and implications based on t-norms and t-conorms, Fuzzy Sets and Systems 11 (1983) 115–134.
[46] I. Bloch, Duality vs. adjunction for fuzzy mathematical morphology and general form of fuzzy erosions and dilations, Fuzzy Sets and Systems 160 (13) (2009) 1858–1867.
[47] A. Gelb, Applied Optimal Estimation, The MIT Press, 1974.
[48] G. Lakemeyer, E. Sklar, D.G. Sorrenti, T. Takahashi (Eds.), RoboCup 2006: Robot Soccer World Cup X, Lecture Notes in Artificial Intelligence, vol. 4434, Springer-Verlag, 2007.
[49] D. Herrero-Pérez, H. Martínez-Barberá, Robust and efficient field features detection for localization, in: Robot Soccer World Cup X, Lecture Notes in Artificial Intelligence, vol. 4434, Bremen, Germany, 2007, pp. 347–354.
[50] V. Nguyen, A. Martinelli, N. Tomatis, R. Siegwart, A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05), Edmonton, Canada, 2005, pp. 1929–1934.
[51] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, John Wiley and Sons, Wiley-Interscience Publication, 1973.
[52] D. Cohen, Y.H. Ooi, P. Vernaza, D.D. Lee, The University of Pennsylvania Robocup 2004 legged soccer team, in: D. Nardi, M. Riedmiller, C. Sammut, J. Santos-Victor (Eds.), Robot Soccer World Cup, LNCS, vol. VIII, Springer, 2005.
[53] A. Doucet, S.J. Godsill, C. Andrieu, On sequential Monte Carlo sampling methods for Bayesian filtering, Statistics and Computing 10 (3) (2000) 197–208.