Uncertainty-based Localization in a Topological Robot Navigation System

Conference Paper · April 2017

DOI: 10.1109/ICARSC.2017.7964054

4 authors:

Clara Gomez
University Carlos III de Madrid

Alejandra Carolina Hernández
University Carlos III de Madrid

Jonathan Crespo
University Carlos III de Madrid

Ramon Barber
University Carlos III de Madrid

2017 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
April 26-28, Coimbra, Portugal

Uncertainty-based Localization in a Topological Robot Navigation System

C. Gomez, A.C. Hernandez, J. Crespo and R. Barber
System Engineering and Automation Department, RoboticsLab
Universidad Carlos III of Madrid, Spain
Email: clgomezb@ing.uc3m.es, alejhern@ing.uc3m.es, jocrespo@ing.uc3m.es, rbarber@ing.uc3m.es

***Abstract*—Localization is the process of knowing and updating continuously a robot position with regards to its environment based on sensor information. Localization strategies are required for accurate mobile robot navigation and they should be adapted to the new tendencies and tools. The main purpose of this work is to develop a localization system that allows a mobile robot to know its position at each moment as well as to identify when it has gotten lost.**

In this paper, localization is applied to a topologically defined environment using Hidden Markov Models (HMMs) as main probability algorithm. HMMs give a stochastic solution applicable to discrete representations such as events associated to sensorial actions. The developed topological localization system requires an a priori environment representation, the acquisition of perceptions related to events, the planning of a path as a sequence of actions and perceptions and the navigation that converts the sequence into real movements.

Finally, experiments have been carried out in a simulated environment, their results show the feasibility of the localization system and motivates the future test in real robotic platforms. The results also encourage to integrate topological and metric information in the probability distributions.

I. Introduction

Navigation has been a main field of study in mobile robotics since its beginning and it is still a challenge. Navigation and mobility have been fundamental in animal development and it is impossible to refer to navigation if localization is not concerned. Animal behaviours for localization are varied and highly developed. From bees and ants localization behaviours based on hormones and tactile signals, to dolphins and bats echolocation techniques, animals localization complexity is remarkable [1] [2].

Depending on the navigation abstraction level, the localization technique is different. In this work, navigation is tackled from topological representations and behaviours so localization algorithms have to be founded on a state structure and applicable to discrete models.

The main motivation of this work is to provide a robot with localization capabilities so it can move naturally in a human environment. As robots get more integrated in human environments, their requirements increase: they have to perceive accurately the environment, move safely and naturally, locate themselves and do not get lost, communicate with humans fluently, etc. These functions have been widely studied but they are not solved yet. As technology improves with more powerful computers and more accurate sensors, new possibilities appear. This work is focused on modelling the uncertainty of navigation in human environments in order to develop a more robust navigation system and provide a robot with the information required to know its position and develop strategies to deal with the situations when it gets lost. Given the imprecisions of sensorial systems, the need for inclusion of complex and high abstraction level arises.

The structure of this paper is as follows: in Section 2, the main works regarding robot localization are included. Section 3 describes the general system overview and concepts. In Section 4, the localization algorithm is explained. Finally, Section 5 and 6 present the experimental results and the conclusions obtained with this work.

II. Related Works

There are some previous works that combine topological representations and localization methods based on HMMs, as presented below.

HMMs have been used for a long time, firstly regarding pattern recognition, specially speech recognition and more recently regarding navigation, localization and place recognition. The mathematics behind HMMs appeared firstly in [3] where they stated the formal definition of the model. In [4] and [5], the theory of Markov Models is explained and they focus on applying them to speech recognition systems. Other works such as [6] and [7] use HMM to recognize distinctive places in indoor environments. Regarding navigation and localization applications, [8] stated an experimental comparison of localization methods. They compared HMMs and Kalman Filter methods applied to metric representations of the environment. They concluded that methods based on Kalman Filter were slightly more efficient, while methods based on HMM were more robust. In [9] and [10], they used Markov Models to model navigation uncertainties. Other authors [11], used HMMs to acquire a topological model from visual sensing, establishing the spatial relationships between locations.

Regarding robot localization many different approaches have been studied. Most popular approach is Simultaneous Localization And Mapping (SLAM). SLAM has been mainly studied from a metric point of view [12] [13]. Other authors have used landmarks to improve SLAM techniques [14] and other authors have used SLAM to localize a robot in hybrid maps, global topological representation and local metric maps

[15]. But localization has not only been studied from a SLAM point of view, researches have also focused on different localization techniques. In [16], they use Kalman Filters to track 3D landmarks and localize the robot globally from the current view of the scene. In [17], the environment is represented as a topological model based on a Voronoi Diagram and localization is performed using qualitative information associated with the nodes of the map. Other authors have used probabilistic Bayesian methods, as Markov Models, to localize a robot in a topological representation [18] and [19].

This work offers a different approach to model the uncertainty in a topological navigation system and proposes a localization implementation based on perceptual structures.

III. SYSTEM OVERVIEW

In this work, a system to localize robots in an environment has been designed. The purpose of this system is to guarantee that a robot locates itself robustly and effectively and to recover and recalculate its position if it has gotten lost. Currently, the navigation system implemented for the robot is topological [20] [21], so the localization algorithm has to be integrated in the global topological structure and oriented to work with discrete representations. The localization system was developed using ROS, Robot Operating System, in order to enhance the integration between modules.

In this case, the localization system integrates: Map acquisition module, Path planning module, Perception module and Localization Algorithm module as shown in Figure 1.

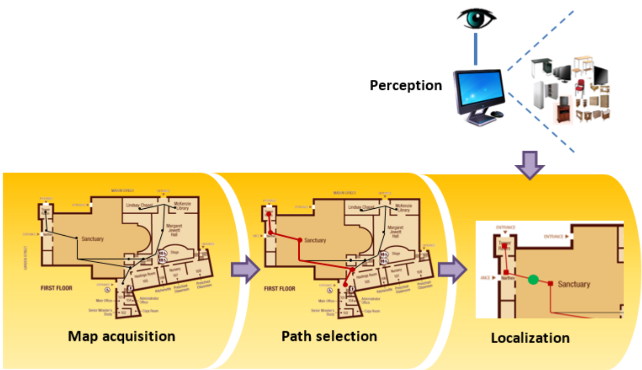

Fig. 1. General representation of the system: Map acquisition, path planning, localization and perception

Map is built a priori and stored in a database. Pure topological maps contain information regarding nodes and their connections, arches, associating nodes with their perceptual events and arches with their abilities. Maps can contain extra information about nodes and arches as travelling distance, heading orientation, etc. In this case, this information is stored as parameters. In Figure 2 metric representation of the environment and its topological abstraction are shown. At each planning and localization stage, this representation is read from the database. The information contained in the map is

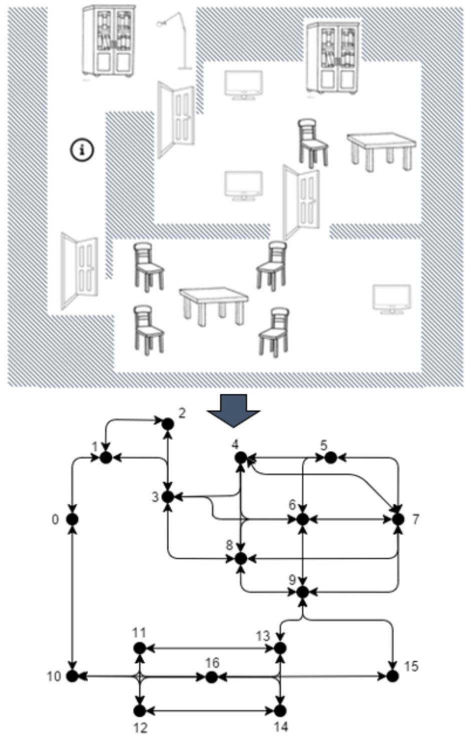

Fig. 2. Geometric map representation of the environment and topological abstraction

used for the path-planning and lately for the generation of the probability distributions at the localization step.

Planning is managed from a topological point of view, the plan corresponds with the succession of events and abilities that the robot will have to fulfil. Dijkstra algorithm [22], which is a graph path-planning algorithm, was chosen to plan the trajectory. Dijkstra algorithm is designed in order to obtain the shortest path between nodes in a graph. Given an initial node it evaluates adjacent nodes, giving priority to the ones with shorter distance, and iterates until the goal node is reached or every connection between nodes has been explored. The topological planning algorithm implemented in this work is a modified Dijkstra, in which the cost of a translation can be varied modifying the cost function with feedback of previous executions, such as a personality factor [23].

Several perception modules have been integrated in a Perception Interface. In this case Initial Odometry Position, Chairs, Tables, Closets, Screens, Lamps and Doors are the perceptions included in the environment. This interface outputs a message containing every possible perception and its observation probability, as shown in Figure 3. Computer vision and object detection systems are not robust enough to detect an object at every condition so the uncertainty has to be modelled and introduced in further algorithms.

Taking into account the map, the path and the observations received the localization algorithm is able to compute robot's position. An stochastic localization algorithm is implemented for this work. It is based on pure topological concepts so its

2017 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
April 26-28, Coimbra, Portugal

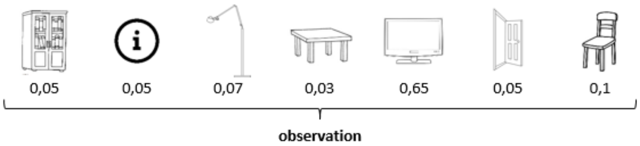

Fig. 3. Perception uncertainty representation

output is determined by the current state (node) of the robot and the perceived observations. The localization algorithm is explained in detail in Section IV.

IV. Localization algorithm based on HMMs

A localization algorithm based on HMMs is proposed in this paper. Localization is mandatory in order to build a robust navigation system, the robot needs to know its position in order to establish the correct relations with its environment. In this case, localization is applied on top of a pure topological representation of the environment so the localization algorithm will determine the probability of being at each node of the map. This type of behaviour corresponds to a discrete algorithm. Although HMM have been widely studied, it is one of the most robust algorithms to manage discrete data probabilities.

Markov model is a stochastic model used to characterize systems where it is assumed that future states depend only on the current state, not on the events that occurred before. A Hidden Markov model (HMM) is a Markov model in which the system being modelled is assumed to have unobserved, hidden, states.

HMMs are represented using the compact notation λ, equation 1. The model parameters that characterize every HMM are A, the state transition probability distribution, B, the observation probability distribution, and, π the initial state probability distribution.

$$\lambda = \{A, B, \pi\} \tag{1}$$

In this work, we have reduced the problem of HMMs to a simple forward procedure. Forward probabilities α are calculated as the probability of being at state *i* given the observation *b*. Forward procedure is iterative and new probabilities are calculated at each time *t*. In equation 2, the expression for calculating the forward probability corresponding to the first observation, $t = 0$, is presented. In equation 3, the expression for calculating forward probabilities corresponding to later observations, $t > 0$, is presented. In equations 2 and 3, Q represents the state at time t, S denotes the possible individual states and O represents the observation.

$$\begin{aligned} \alpha_0(i) &= \frac{P(Q_i|S_i) * P(O_0 = b_0|Q_i = S_i)}{\sum_{j=s}^{N} \alpha_0(i)} \\ &= \frac{\pi_0 * P(b_0|S_i)}{\sum_{j=s}^{N} \alpha_0(i)} \end{aligned} \tag{2}$$

$$\alpha_t(i) = \frac{(\sum_{j=t}^{N} \alpha_{t-1}(j) * a_{j,i}) * P(b_t|S_i)}{\sum_{j=s}^{N} \alpha_0(i)} \tag{3}$$

In order to obtain a total probability distribution of 1 at each forward probability iteration, normalization is required.

A re-estimation procedure is also implemented in order to adjust the transition probability distributions after each observation. The new value for each state transition, equation 4, is calculated from every previous forward probability, its assigned state transition probability and the probability of receiving the observation that occurred in that state. Similarly to the calculation of forward probabilities, normalization is required after re-estimation.

$$NewA_{i,j} = \frac{\sum_{j=t}^{N} \alpha_t(i) * A_{i,j} * P(O_{t+1}|S_j)}{\sum_{j=t}^{N} \alpha_t(i)} \tag{4}$$

This mathematical interpretation of HMMs based on forward procedures has been modelled according to the flowchart shown in Figure 4. Firstly, the map is acquired from the environment model and the user sets the initial and goal nodes for the robot. Then, the path-planning algorithm determines the optimum path. Once the map and the path are available the localization algorithm can start by initializing the parameters of the model. After receiving the first observation, forward probability distribution is calculated and the most likely state is published. When new observations arrive, the re-estimation procedure is executed and new forward probabilities distributions are computed publishing the most likely state in each iteration.

Regarding the parameter initialization some assumptions have been made and some dependencies have been established. As the map and the events (observations) associated with each node of the map are known a priori, we can assume that the number of states or nodes, *n*, and the number of types of observations, *m*, are known. These assumptions determine the size of the probabilities matrices, represented in equations 5, 6 and 7.

$$\pi = \begin{pmatrix} \pi_1 & \pi_2 & ... & \pi_n \end{pmatrix} \tag{5}$$

$$B = \begin{pmatrix} b_{0,s=0} & b_{1,s=0} & \cdots & b_{m,s=0} \\ b_{0,s=1} & b_{1,s=1} & \cdots & b_{m,s=1} \\ \vdots & \vdots & \ddots & \vdots \\ b_{0,s=n} & b_{1,s=n} & \cdots & b_{m,s=n} \end{pmatrix} \tag{6}$$

$$A = \begin{pmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,n} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,0} & a_{n,1} & \cdots & a_{n,n} \end{pmatrix} \tag{7}$$

π matrix values correspond to the initial state probabilities, the value that corresponds with the initial node the user has commanded is set to 0.8, while the others are calculated according to $\frac{0.2}{n-1}$. In B matrix, columns correspond to different

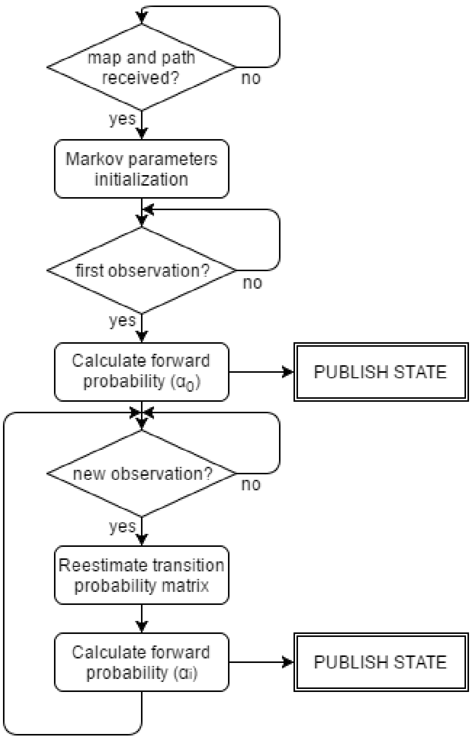

Fig. 4. Flowchart for localization algorithm

observations and rows correspond to states. So each value indicates the probability of perceiving an observation in a state. Dependencies regarding B matrix are established according to the observation associated with each node of the map, in each state the probability value of perceiving the corresponding observation is increased in relation with the other values. A matrix entries correspond to the probability of translating from one state to another. The values which are associated to connections in the map are greater than the ones that associate to nodes that are not connected. Moreover, the values that correspond to translations in the selected path are also increased. The dependencies described above for each model matrix are illustrated in Figure 5.

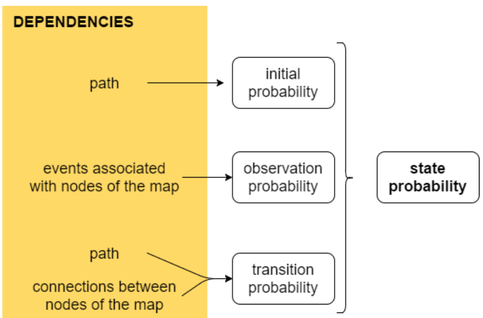

Fig. 5. Dependencies of the probability distributions associated with Markov algorithm

V. Experimental Results

Results were obtained from simulation experiments. Path-planning and localization algorithms were running using simulated observations. Observations were created as a probability distribution among the different possible observations, as shown in Figure 3. For example, one observation can be defined as 80% chair, 15% table and 5% door depending on the information offered by the perception modules. None observation used in these experiments corresponds 100% to one observation type as we try to simulate the uncertainty found in real perception systems.

In order to test the failures of the localization system and its recovery capacity, several errors were introduced in the observations. For each path, the next tests were developed:

- correct observations according to the path.
- missing one observation within the path.
- repeating one observation within the path.
- including one extra observation different from the ones required.

Experiments shown include a simple path representation and a more ambiguous path representation. The former was simulated with and without re-estimation of the transition matrix. Results show the differences and the improvement due to the re-estimation procedure.

A. *Simple path localization experiment*

In this experiment the system is asked to compute the path from node 0 to node 7. The optimum trajectory in this case is 0-1-3-4-7. Node 0 corresponds to the initial position, 1 corresponds to a closet, 3 to a door, 4 to a screen and finally 7 to a table. This experiment is considered simple as the final observation, table, is only present in this environment twice and far from each other.

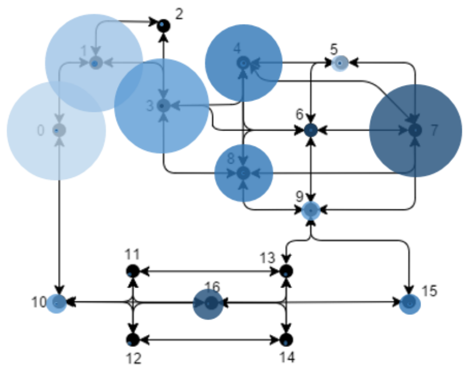

Fig. 6. Probability result for the trajectory from node 0 to node 7 when correctly executed.

In Figure 6, the resulting probability distribution for this path when the observations are correct (same observation as expected regarding the map) is shown. The size of the bubbles illustrates the probability for each node and the colour of the bubbles gets darker while the iterations increase. Light blue corresponds to the initial node probability while dark blue corresponds to the goal node probability. Exact probability values are shown in Table I.

TABLE I
FORWARD PROBABILITY FOR PATH FROM NODE 0 TO NODE 7 WHEN CORRECTLY EXECUTED

state	α_0	α_1	α_2	α_3	α_4
0	**0.98538**	0.00006	0.00158	0.00020	0.00016
1	0.00234	**0.96892**	0.00015	0.00356	0.00020
2	0.00071	0.00030	0.00158	0.00070	0.00016
3	0.00071	0.00007	**0.90327**	0.00044	0.00191
4	0.00071	0.00006	0.00430	**0.60161**	0.00089
5	0.00234	0.02887	0.00014	0.00048	0.00104
6	0.00071	0.00007	0.00101	0.00076	0.01843
7	0.00071	0.00007	0.00022	0.00012	**0.87200**
8	0.00071	0.00005	0.00214	0.34645	0.00087
9	0.00071	0.00006	0.04063	0.00037	0.00097
10	0.00071	0.00101	0.04062	0.00031	0.00017
11	0.00071	0.00006	0.00051	0.00021	0.00171
12	0.00071	0.00006	0.00051	0.00021	0.00171
13	0.00071	0.00006	0.00050	0.00010	0.00245
14	0.00071	0.00006	0.00050	0.00006	0.00245
15	0.00071	0.00006	0.00214	0.04412	0.00085
16	0.00071	0.00007	0.00011	0.00021	0.09395

The analysis of the results obtained in the different tests is shown in Table II. Apart from the correct execution, tests with errors have been studied. Tests included are *jump*, when the perception module misses the observation of a state, *repetition*, when an observation is processed twice and *inclusion*, when an extra random observation is included. Although there were some errors in middle nodes, the system recovered from this errors successfully reaching always the correct goal node. In order to get an execution unable to recover, the perception system would need to miss two or more observations.

TABLE II
PROBABILITY RESULTS FOR PATH FROM NODE 0 TO NODE 7

test	% right goal probability	% error middle probability	% recovery
correct	100	0	-
jump	100	25	100
repetition	100	12,5	100
inclusion	100	16,67	100
global	100	13,54	100

B. Re-estimation of the transition matrix experiment

In order to study the effect of the re-estimation procedure applied to the localization algorithm, the trajectory from node 11 to node 8 was performed with and without re-estimation. The optimum trajectory in this case is 11-13-9-8. Nodes 11 and 13 are chairs, 9 corresponds to a door and 8 corresponds to a screen. This experiment is more ambiguous as there are several chairs, door and screen close to the path, which can lead the localization algorithm to erroneous conclusions. In Figure 7 and 8, the probability distribution is represented for the trajectory executed without re-estimation and with re-estimation, respectively. As in the previous experiment, the size of the bubbles illustrates the probability for each node and the colour of the bubbles gets darker while the iterations increase. In table III, the probabilities for the correct nodes are compared and the improvement rate of the re-estimation is calculated, giving an average improvement value of 17,48%. As re-estimation is applied after the first iteration of observations, this comparison is not applicable for the first forward probability.

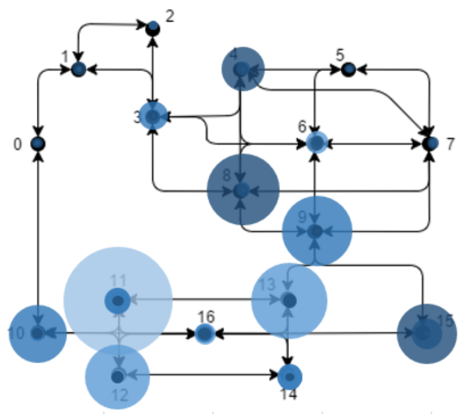

Fig. 7. Probability result for the trajectory from node 11 to node 8 when correctly executed, without re-estimation of the transition matrix

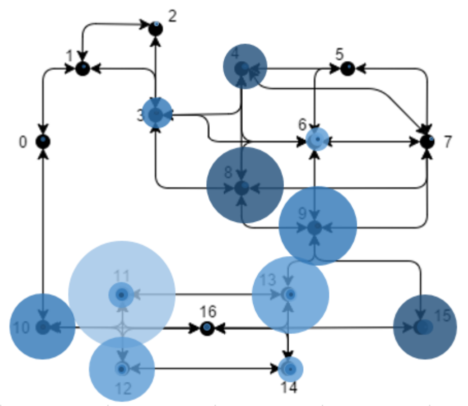

Fig. 8. Probability result for the trajectory from node 11 to node 8 when correctly executed, with re-estimation of the transition matrix

TABLE III
PROBABILITY RESULTS FOR THE CORRECT NODES OF PATH FROM NODE 11 TO NODE 8 WHEN CORRECTLY EXECUTED. COMPARISON BETWEEN METHOD WITH AND WITHOUT RE-ESTIMATION

test	α_0	α_1	α_2	α_3
without re-estimate	0,9301	0,4536	0,3911	0,4071
with re-estimate	0,9301	0,4874	0,5004	0,4765
improvement	-	7,45%	27,95%	17,06%

Finally, the analysis of the results obtained in the different tests is shown in Table IV. Although there were some errors in middle nodes, most of the times the system recovered from these errors successfully. Within the times there were middle node errors, the system recovered 77,78%, giving a global goal reaching of 95,55%. As the path contains more ambiguities, the robustness of the localization algorithm decreases.

TABLE IV
PROBABILITY RESULTS FOR PATH FROM NODE 11 TO NODE 8

test	% right goal probability	% error middle probability	% recovery
correct	100	0	-
jump	88,89	33,33	66,67
repetition	100	20	100
inclusion	93,33	20	66,67
global	95,55	18,33	77,78

VI. CONCLUSION

The aim of this work was to study localization methods applicable to topological representations of the environment and develop a robust localization algorithm capable of updating the position of the robot correctly and recovering in case it gets lost. Simulation experiments were developed using a complex topological map and the results obtained offered positive conclusions.

- 97,7% of the tests reached the correct goal. Although tests included intentional errors in the observations, the system could mostly reach the goal.
- The system was able to recover most of the times when it had gotten lost. Results show a recovery of 88,89% after localization errors.
- The observation error that caused greater failure in localization was missing an observation. This will be taken into account in order to improve the perception system.

Although the results are positive, the system needs to take into account metric conditions of the environment, orientation and distance. Future works include the addition of this information to the pure topological localization algorithm that has been developed in this work. It is assumed that adding this information will solve some of the ambiguities found in the experiments. Finally, once the algorithm is improve, it will be implemented in a real robotic platform [20].

This work represents the uncertainties present in a real navigation task in order to develop a more robust navigation system able to drive a robot in a human environment. As technology improves with more powerful computers and more accurate sensors, new possibilities appear to solve navigation and localization problems.

ACKNOWLEDGMENT

The research leading to these results has received funding from the RoboCity2030-III-CM project (Robótica aplicada a la mejora de la calidad de vida de los ciudadanos. fase III; S2013/MIT-2748), funded by Programas de Actividades I+D en la Comunidad de Madrid and cofunded by Structural Funds of the EU and NAVEGASE-AUTOCOGNAV project (DPI2014-53525-C3-3-R), funded by Ministerio de Economía y Competitividad of SPAIN.

REFERENCES

[1] J. A. Simmons *Directional hearing and sound localization in echolocating animals*. In Directional Hearing (pp. 214-225). Springer US. 1987.
[2] L. Patane, S. Hellbach, A. F. Krause, P. Arena and V. Durr *An insect-inspired bionic sensor for tactile localization and material classification with state-dependent modulation*. Active Touch Sensing, 139. 2014.
[3] L. E. Baum and T. Petrie, *Statistical inference for probabilistic functions of finite state Markov chains*. The annals of mathematical statistics, 37(6), 1554-1563. 1966.
[4] L. R. Rabiner and B. H. Juang, *An Introduction to Hidden Markov Models*. IEEE ASSP Magazine, 3(1), 4-16. 1986.
[5] L. R. Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. Proceedings of the IEEE, 77(2), 257-286. 1989.
[6] O. Aycard, F. Charpillet, D. Fohr and J. F. Mari *Place learning and recognition using hidden markov models*. In Intelligent Robots and Systems, 1997. IROS'97., Proceedings of the 1997 IEEE/RSJ International Conference on (Vol. 3, pp. 1741-1747). 1997.
[7] B. L. Boada, D. Blanco and L. Moreno *Symbolic place recognition in voronoi-based maps by using hidden markov models*. Journal of Intelligent and Robotic Systems, 39(2), 173-197. 2004.
[8] J. S. Gutmann, W. Burgard, D. Fox and K. Konolige *An experimental comparison of localization methods*. In Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on (Vol. 2, pp. 736-743). 1998.
[9] A. R. Cassandra, L. P. Kaelbling and J. A. Kurien *Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation*. In Intelligent Robots and Systems, 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on (Vol. 2, pp. 963-972). 1996.
[10] S. Koenig and R. G. Simmons *Unsupervised learning of probabilistic models for robot navigation*. In Robotics and Automation.Proceedings., 1996 IEEE International Conference on (Vol. 3, pp. 2301-2308). 1996.
[11] J. Kosecka and F. Li *Vision based topological Markov localization*. In Robotics and Automation, 2004. Proceedings. ICRA'04, IEEE International Conference on (Vol. 2, pp. 1481-1486). 2004.
[12] J. J. Leonard and H. F. Durrant-Whyte *Simultaneous map building and localization for an autonomous mobile robot*. In Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on (pp. 1442-1447). 1991.
[13] R. Claessens, Y. Muller and B. Schnieders *Graph-based Simultaneous Localization and Mapping on the TurtleBot platform*. 2013.
[14] S. Thrun, W. Burgard and D. Fox *A probabilistic approach to concurrent mapping and localization for mobile robots*. Autonomous Robots, 5(3-4), 253-271. 1998.
[15] N. Tomatis, I. Nourbakhsh and R. Siegwart *Hybrid simultaneous localization and map building: a natural integration of topological and metric*. Robotics and Autonomous systems, 44(1), 3-14. 2003.
[16] S. Se, D. Lowe and J. Little *Local and global localization for mobile robots using visual landmarks*. In Intelligent Robots and Systems, 2001. Proceedings. IEEE/RSJ Int Conf on (Vol. 1, pp. 414-420). 2001.
[17] P. Ranganathan, J. B. Hayet, M. Devy, S. Hutchinson and F. Lerasle *Topological navigation and qualitative localization for indoor environment using multi-sensory perception*. Robotics and Autonomous systems, 41(2), 137-144. 2002.
[18] F. Dayoub and T. Duckett *An adaptive appearance-based map for long-term topological localization of mobile robots*. In 2008 IEEE/RSJ Int Conf on Intelligent Robots and Systems (pp. 3364-3369). 2008.
[19] H. Cheng, H. Chen and Y. Liu *Topological Indoor Localization and Navigation for Autonomous Mobile Robot*. IEEE Transactions on Automation Science and Engineering, 12(2), 729-738. 2015.
[20] C. Gomez, A. C. Hernandez, J. Crespo and R. Barber *Integration of Multiple Events in a Topological Autonomous Navigation System*. In Autonomous Robot Systems and Competitions (ICARSC), Int Conf on (pp. 41-46). 2016.
[21] A. C. Hernandez, C. Gomez, J. Crespo and R. Barber *Object Classification in Natural Environments for Mobile Robot Navigation*. In Autonomous Robot Systems and Competitions (ICARSC), Int Conf on (pp. 217-222). 2016.
[22] Skiena S., *Dijkstra's Algorithm. Implementing Discrete Mathematics* Addison-Wesley, 225-227., 1990.
[23] V. Egido, R. Barber, M. J. L. Boada and M. A. Salichs, *A planner for topological navigation based on previous experiences.* Parameters, 2, 3, 2004.