

Unified Fusion System Based on Bayesian Networks for Autonomous Mobile Robots

Eva Besada-Portas

Jose A. Lopez-Orozco

Jesus M. de la Cruz

Departamento de Arquitectura de Computadores y Automatica
Fac. CC. Fisicas. Universidad Complutense Madrid
28040 Madrid, Spain
 {evabes,jalo,jmcruz}@dacya.ucm.es

Abstract – *A multisensor fusion system that is used for estimating the location of a robot and the state of the objects around is presented. The whole fusion system has been implemented as a Dynamic Bayesian Networks (DBN) with the purpose of having a homogenous and formalized way of capturing the dependencies that exist between the robot location, the state of the environment, and all the sensorial data. At this stage of the research it consists of two independent DBNs, one for estimating the robot location and another for building an occupancy probabilistic map of the environment, which are the basis of a unified fusion system. The dependencies of the variables and information in the two DBN will be captured by a unique DBN constructed by adding arcs (and nodes if necessary) between the two DBN. The DBN implemented so far can be used in robots with different sets of sensors.*

Keywords: multisensor fusion system, bayesian networks, autonomous mobile robots.

1 Introduction

Autonomous mobile robots working in a changing and/or unknown environment need a perception system to know their position and what is happening around. The perception system is in charge of obtaining the information provided by all the sensors and process it to estimate the robot location and the state of its environment. The estimations are used in the robot control system to act according to its original purpose and the observed changes. Therefore the perception system needs to obtain results as quick as possible to let the robot take real time decisions.

Mobile robot designers use multiple and different types of sensors due to the advantages and limitation of each type. Cost, precision, range, scan rate and type of information are factors that influence on the decisions of the designers. The limitations of each type are usually solved using redundant and/or complementary sensors. Redundant information reduces the uncertainty of the sensorial information and makes the system robust to the failure and malfunction of some sensors. Complementary information allows to observe a bigger set of

characteristics. The use of multiple information sources is advantageous [1] because: it increments the reliability of the perception system, it favors the observation of multiple characteristics, and it increments the speed of the system while decrementing its cost.

The information from the different sources is processed and merged in the multisensor fusion subsystem of the perception system. The final goal of the fusion subsystem is to obtain the robot location and some information (for instance a map) about the state and position of the objects around it. Many multisensor fusion algorithms for solving the perception problem of mobile robots can be found in the literature [1,2]. Lots of them are especially implemented for a robot and so they can not be easily reused in other robots.

The robotic fusion system is usually implemented with two separated subsystems, one in charge of estimating the robot position (location subsystem) and the other the state of its surroundings (map building subsystem). In some specific cases a unique system is used, although the estimation is usually performed in two stages [3, 4]. Moreover, the location and map building fusion subsystems are usually implemented using different fusion techniques. This division is not natural because, for instance, when building the surrounding map the robot location is usually needed (especially when there is an initial map) and information about the environment can be useful for improving the location estimation. The dependency between the two types of information can be captured easier if a unified fusion system is used than when the system is divided in two parts that are allowed to exchange information. In a unified system the information about the robot and objects locations can be fused at the same level and the influence of the uncertainty of the robot and surroundings merged easier. In a divided system the information of each subsystem is used, not fused, in the other subsystem and so it is more difficult to capture the underlying dependencies between all the sensor data. The difficulties are bigger when each fusion subsystem uses a different fusion technique and so the information returned by each has different characteristics.

This paper presents a unified multisensor fusion system for autonomous mobile robots based on Dynamic Bayesian Networks (DBN) [5,6] which estimates the robot location and constructs a dynamic map of the environment (using if provided a initial map). The designed system is versatile and can be reconfigured for different robots and/or different types of sensors, after providing some information about the sensors.

At this stage of the research the fusion system consists on two independent DBN, one for the robot location and another for the map. Although the fusion system is still divided, it is already unified by the use of the same technique. The dependencies of the variables in the two DBN will be captured by a unique DBN, built by adding arcs and nodes between the two DBN.

The paper is organized in the following way. In section 2 there is a description of Dynamic Bayesian Networks which highlights the properties that make them useful in multisensor fusion systems. In section 3 the DBN of our fusion system are presented and explained in detail. Section 4 presents experimental results obtained when the fusion system is used in a real time control application for autonomous mobile robots. Finally, in the last section some conclusions are presented.

2 Dynamic Bayesian Networks and Fusion Systems

Bayesian Networks (BN) are graphical models that provide a probabilistic and graphical framework for dealing with the complexity and uncertainty inherent to many probabilistic problems. Their graphical side is itself an appealing interface for modeling complex relations between sets of random variables. The probability theory behind ensures that the system is consistent and it provides inference mechanisms for obtaining the probabilities of subsets of random variables and propagating through the model the evidence of the data.

BNs are directed acyclic graphs whose nodes are random variables (discrete or continuous), whose structure (arcs) contains independence assumptions between nodes, and which are parameterized by the conditional probabilities needed for specifying the underlying distribution (the probability of a node given its parents). The evidence (knowledge about the state of a node) can be entered in any node and propagated to update the probabilities through all the network using an inference mechanism. There are many inference engines specially designed for different types of BNs, either for exact inference and for approximate inference. When exact propagation is performed, the final probability distribution given all the evidence does not depend on the order in which the evidence is entered in the BN. Approximate inference is usually used in complex BNs

where exact inference is not efficient and it is necessary in BNs for which exact methods have not been developed. Additionally more tasks than inference can be performed: finding the most probable configuration, detecting conflicts in the evidence, learning ...

Dynamic Bayesian Networks (DBN) are a series of BN which is expanded over time. The different BNs (called network slices) are linked by arcs that are in charge of capturing the evolution of the probabilities of the variables with time. The arcs between slices are similar to the arcs between the nodes inside a slice, so the DBN is itself a BN. Therefore the same properties and inference engines can be used. The expansion of the DBN can be due to different causes such as the end of a waiting period or the observation of an external event. The different networks slices don't need to have the same structure and to reduce the memory and inference costs, the network is pruned cyclically and only the slices that fit in a window around the current time are maintained.

The use of BN and DBN in data fusion and related tasks is not new and several examples can be found in the literature: BNs for validating the sensorial information in order to detect malfunctioning sensors [7]; BNs for detecting speakers and which integrates different types of information extracted from a vision system [8]; BNs for tracking discontinuous motion which fuses high-level contextual information with sensor-level information [9]; BNs for building an occupancy map of the environment [10], discrete DBNs for estimating the position of a mobile in environments divided in few cells [11, 12]; continuous DBN which are equivalent to Kalman and smoothing filters [13]; ... In the examples BN and DBN are used for dealing with different parts of the data fusion problem. A whole fusion system, with different functionality levels can be implemented using several ideas present in the cited references.

The design of a fusion system is especially favored for the structure of DBNs and its graphical interface. The world can be modeled initially with a simple DBN, which can be easily modified for including more sensorial information and more complex relations between the characteristic of the world (by adding nodes and arcs and specifying new conditional probabilities). In this way the functionality of the system can be easily incremented. Additionally, the layered structure of the BNs lead us naturally to a fusion system with different levels where the flux of information is controlled by the propagating scheme used in the BN. The information in the low levels will influence the state of the higher and vice versa.

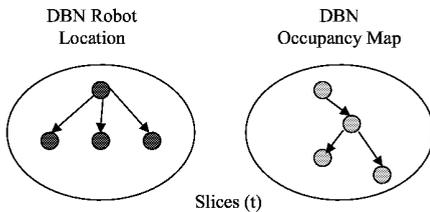
The use of DBNs for the whole fusion system in the autonomous mobile robot problem, will let us build, in a next stage, a unified fusion system where the uncertainty about the position and map influences each other.

3 Towards a Unified Fusion System

In this section a general multisensor fusion system based on DBNs and designed for autonomous mobile robots with several sensors, which is in charge of estimating the robot location and the state of the environment, is presented. Here, we use *general* in the sense that the system can be configured by the user for different robots and different sets of sensors. The fusion system needs to have the sensorial information provided in the correct format and a set of parameters need to be specified.

The fusion system is considered unified because it has been implemented using a unique technique (DBN) with the purpose of having a homogenous and formalized way of capturing better the dependencies between the estimation of the robot location and the estimation of a map (which stores the state of the environment). In the current stage of the research, our attention is more focused in exploiting the sensorial information about the robot location and map subsystems (which are next explained) than in analyzing the dependencies among them. So the designed DBN is not connected. The two fusion systems are probabilistic independent because there are not arcs between the nodes in charge of modeling the sensorial and fusion information of the robot location and those of the map. In a second stage we will joint the two subsystems, adding arcs (and nodes if necessary), so the dependencies will be captured by the DBN. Figure 1 schematizes both cases, at the top the unified subsystem when the two fusion subsystems are independent because there are not arcs between them; and at the bottom the unified system with dependent fusion subsystems.

a) Unified Fusion System with independent Fusion Subsystems



b) Unified Fusion System with dependent Fusion Subsystems

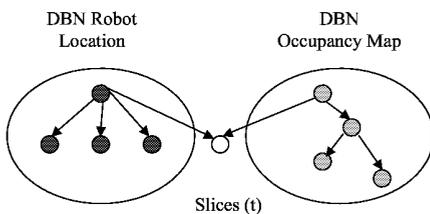


Figure 1. Unified fusion systems with independent and dependent fusion subsystems.

3.1 Location Subsystem

The location subsystem of the unified fusion system is in charge of estimating the current position of the robot given the measures of the position sensors of the robot. In a previous step of the research we implemented an asynchronous distributed Kalman filter for this purpose [14]. The DBN in figure 2, whose nodes are gaussian continuous variables, can be used for modeling the temporal behavior of dynamic linear systems [13]. The robot location is estimated in the $x(t_i)$ nodes while the sensor information is estimated and provided through the $z(t_i)$ nodes.

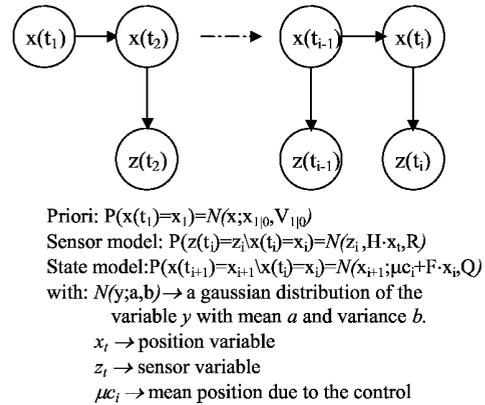


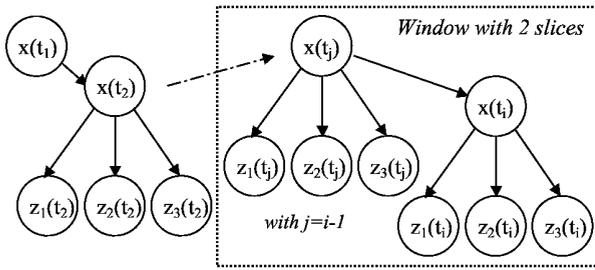
Figure 2. DBN for modeling a dynamic linear system.

The exact inference mechanism over the junction tree for continuous gaussian BN shows that when the root is considered the last location state, the collect evidence step of the inference implements the classical Kalman filter while the distribute evidence step implements a smoothing filter, both filters in the information form. So, the inference engine propagates the evidence in the two directions and each location node has an estimation of the position given all the evidence. In order to reduce the overload of the subsystem due to smoothing calculations, the distribute evidence step can be disabled.

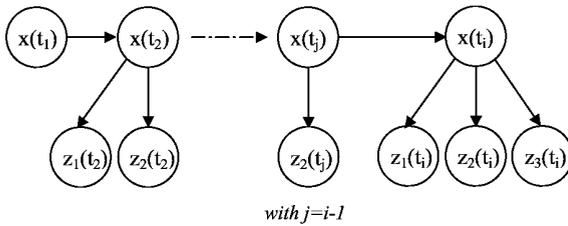
In a mobile robot there are usually several position sensors which provide data at different speeds. When the noises of the different sensors are uncorrelated, the observation model can be partitioned and the distributed version of the Information/Kalman filter [15] can be used. The distributed filter is advantageous in case of having sensors providing measures at different rates.

The location fusion subsystem for asynchronous measurements can also be implemented with DBNs. Figure 3 shows two equivalent solutions. Both of them implement the distributed information filter when only the collect evidence step is performed.

a) DBN with sensor nodes for all the sensors in every time slice



b) DBN with nodes for the sensors with provide a measure in that time slice



In both cases:

$$P(x(t_1)=x_1)=N(x_1; x_{10}, V_{10})$$

$$P(z_j(t_i)=z_{ji} | x(t_i)=x_i) = N(z_{ji}; H_j, x_i, R_j)$$

$$P(x(t_{i+1})=x_{i+1} | x(t_i)=x_i) = N(x_{i+1}; \mu c_i + F \cdot x_i, Q)$$

With: $N(y; a, b) \rightarrow$ a gaussian distribution of the variable y with mean a and variance b .

$x_i \rightarrow$ position variable at time t

$z_{ji} \rightarrow$ sensor variable for the j sensor at time t

$\mu c_i \rightarrow$ mean position due to the control

Figure 3. DBN for modeling dynamic linear systems with several asynchronous sensors.

In both examples the location of the robot ($x(t_i)$) is estimated with the help of three sensors, which provide data asynchronously. When necessary, more sensors nodes can be used. In figure 3, the sensor nodes are shaded if the corresponding sensor has provided a measure at that time and they are not white if not measure has been provided. In the first case (figure 3a), the structure of the DBN is the same in every time slice, in spite of the fact that not all the sensors are not providing data for all the time slices. If there is not data for a node in a time slice, no evidence is entered in that node. So, the existence of nodes without sensing information does not change the estimation of the state position. However the existence of the sensing nodes without information can cause a lose of efficiency in the inference engine. In the second case (figure 3b), the structure of the DBN changes in time according with the data that will be provided at each time slice. With the second DBN the overload in the inference engine due to useless nodes is avoided. However expanding the network according with the sensorial data that will exist in the next time slice is more difficult.

Another aspect to take into account is the possible existence of communication delays in the fusion system.

The approach implemented in our system to deal with small delays is waiting a small time (activation period) before performing each slice expansion to favor the arrival of the sensorial information for that instant in time. If the delay is bigger the data can also be used if the measure falls inside the window around the current time slice. When data is delayed more than the activation period the structure of the first DBN is advantageous. The node for the sensor at that time already exist and so it is enough when incorporating the data in the corresponding node and propagate the evidence. As the final estimation does not depend in the order in which the evidence is introduced in the network, the current state will be equal to the state obtained if the data had arrived on time to the network. The size of the window should be selected so the delayed data usually falls inside it.

When working with communication delays the DBN is better than the distributed Kalman/Information filter because only the delayed evidence need to be introduced in the network and propagated. The other evidence has been already stored in the network and propagated.

Another advantageous aspect of the usage of DBN for the location fusion system is that the variables don't need to be gaussian with linear dependencies. Other probabilities distribution can be used and propagated, using an appropriated inference engine.

The final implemented location fusion system implemented so far let users decide the kind of DBN (a or b) to use, and in the first case the size of the window. Additionally, users should provide the linear state and sensor models, the covariance of those models and the speed rate of each sensor. Other probabilities distribution and dynamic models could also be used selecting an appropriated inference engine.

3.2 Map Building Subsystem

The map fusion subsystem is in charge of estimating the state of the objects around the robot. The information of the state of the environment can be stored in different formats, being a multidimensional grid where each cell corresponds to a region of the environment a well-known approach. A BN can be used for building a map [4, 10], where the dependency relations between the variables that describe the state of the environment and those that are used for entering the sensorial evidence are caught by the structure of the BN. When many variables are used to describe the environment in detail, the complexity of the problem is increased and so the BN more difficult to obtain. A basic occupancy bidimensional (map) can be obtained using the discrete DBN in figure 4. The state of cell (x) can be occupied or empty, and for each of the sensors (z_j) the possible states are detected and not detected. The BN of the figure is for 3 sensor capable of

detecting objects. More sensors can be added or more complex BN implemented. Sensor models store the detection probability $P(\text{detected}|\text{occupied})$ and the false alarm probability $P(\text{detected}|\text{empty})$ of each sensor, which in this case depend on the distance between the mobile robot and the cell.

Discrete DBN (new nodes for incorporating soft evidence are added when there is new data)

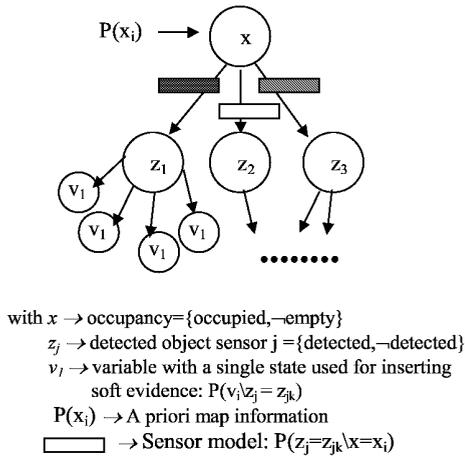


Figure 4. Discrete DBN for modeling the fusion of environment measures in a cell of the map.

The BN is dynamic because its structure changes over time. The sensorial information is inserted in the network as soft evidence. For each new data about a cell, a new v_i node is connected to the sensor node of the sensor that provides the data and the belief probability obtained by the sensor for each of the states of the sensor node (detected, not detected) used as the model between the sensor node and v_i . Then the inference engine is used for propagating new evidence through the network.

The fusion algorithm implemented with this DBN does not work properly if the sensors provide information only about the presence of objects (and not about the absence of objects). The problem arises if in an empty cell an object was detected (either because an erroneous measure was taken or the cell was occupied before for an object that is moving) because as there are not measures against the past measures the probability of occupancy of the cell will not be decremented. If the sensors provide also information about the no detection of objects the system will work properly. However, these types of measures overload the inference, especially in open environments where there are more empty cells than occupied cells around the robot.

The use of forgetting factors, that will make the state of a cell be unknown, is an optional solution to avoid the computational overload. The forgetting factor can be implemented with the DBN presented in figure 5. In this

case only 2 sensors are used and the nodes necessary for introducing the soft evidence are not drawn to make the schema more simple. In the proposed DBN the forgetting factor model will make the fusion system forget both the sensorial and a priori information.

Discrete DBN with a forgetting factor model

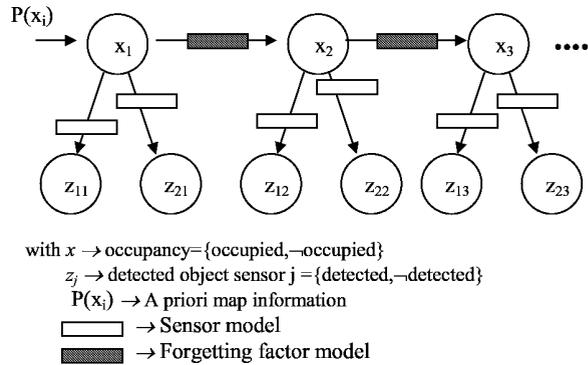


Figure 5. Discrete DBN for the fusion of environment measures in a cell of a map including forgetting factors

The final map building fusion system implemented so far can be used with several sensors and users should provide the sensor model of each sensor. The forgetting model it is only necessary if the sensors don't provide information about the absence of objects in a cell. When forgetting factors are used the user should select the forgetting factor model and the frequency to apply it.

Although the DBN is quite basic and the sensorial information is not fully exploited when sensors provide more information than the one related with the detection of objects, the occupancy grid which is built with it has been used successfully for finding paths in a changing environment. The DBN can be modified to implement a map building fusion system for using other type of sensorial information and building a map which stores more characteristics than each cell occupancy state.

4 Experimental Results

The unified fusion system is embedded in a real time complex program developed for controlling autonomous mobile robots [16]. The application has been tested in different robots with different sets of sensors. The results of two different tests are presented in this paper.

The robot of the examples has three sensors for estimating its location and two sensors for building the map. The location sensors are a magnetic compass (measuring every 1 sec.), encoders (0.5 sec.) and beacon receivers (1.5 sec.). The map sensors are a ultrasonic belt (8 fixed sensors, 3 sec.) and a ultrasonic scanner (2 rotating sensors, 5 sec.).

In the first example the robot is asked to go from (a) to (b), in the laboratory shown in figure 6. The initial position and a priori map (without the two shaded round objects) are provided as a priori information to the fusion system. The back green region is occupied by people moving. The circles in the figure show the estimated robot trajectory, that avoids the two unknown obstacles.

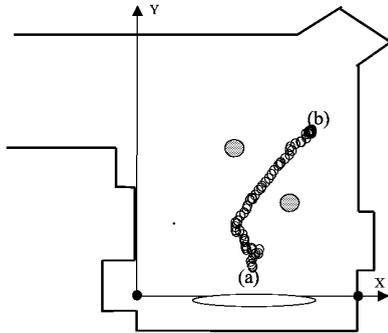


Figure 6. Laboratory and trajectory of the robot

The measures from the position sensors are shown in figure 7: position data at the top and orientation data at the bottom. The position graphic shows the robot trajectory measured by each sensor. The orientation graphic shows the robot orientation at different instants of time.

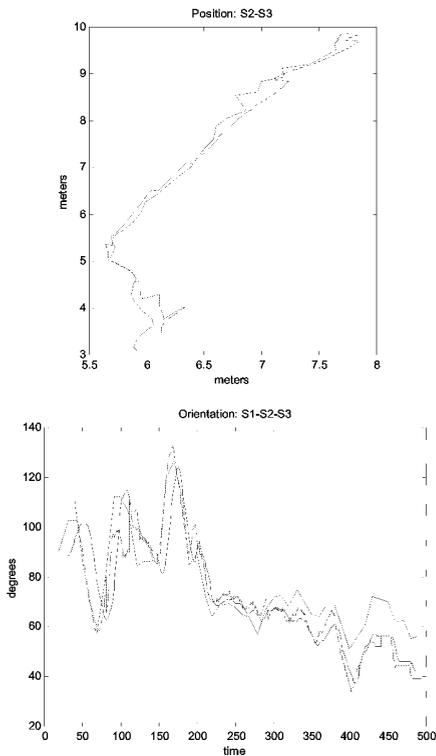


Figure 7. Data from the location sensors

The encoders and beacon receivers provide the two kinds of data while the magnetic compass only provides orientation. The data provided by each sensor differs from the data provided by the others.

The unified location subsystem fuses the data from the three sensors in order to estimate the robot location. The inference engine of the DBN propagates the evidence provided by each sensor through the network. The mean value of the robot location at each time slice can be obtained from the DBN and returned by the location subsystem as the estimation of the robot location. The accuracy of each sensor, specified by the covariance of each sensor model, weights the influence of each sensor in the final location estimation. The estimation of the robot position and orientation are presented in figure 8.

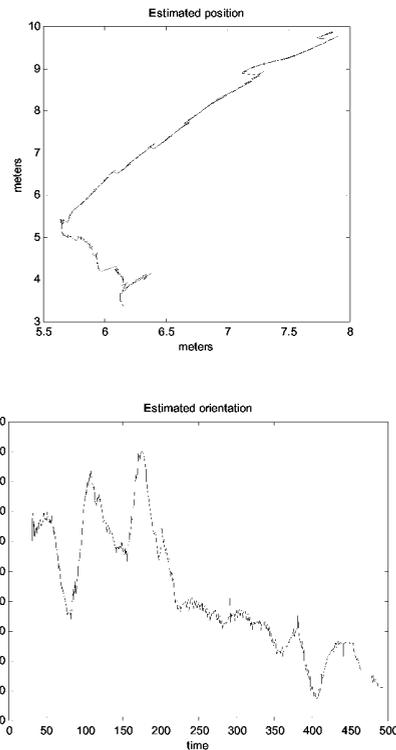


Figure 8. Estimation of the robot location

All the measures from the environment sensors are shown in figure 9. Data from the ultrasonic belt is represented with red circles while data from the ultrasonic scanner is represented with blue *. In this example only evidence about the presence of objects is used. As the ultrasonic belt consists of 8 sensors and measures more frequently than the ultrasonic scanner with 2 sensors, there is more information about the first sensor than about the second. The black line describes the estimated trajectory, which is used with the range data provided by the sonar, in the map building system for deciding in which cells is an object detected.

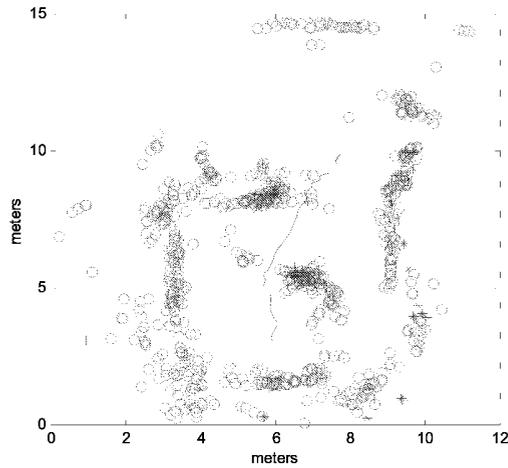


Figure 9. Data from the environment sensors

The final map obtained by the map building subsystem after fusing the a priori information (initial probabilistic map where the a priori probability of being occupied of the cells corresponding to a wall in figure 6 is equal to 0.8 and for the remaining cells is equal to 0.2) with the sensing data during all the robot trajectory is presented in figure 10. Cells with a probability of being occupied lower than .6 are drawn in white. The estimated trajectory is drawn with blue circles. The forgetting model is used with a low forgetting speed and frequency. The forgetting model is applied only in the cells that fall inside a circular area centered in the robot to avoid forgetting information about areas that the robot can not measure. In the final map the areas corresponding to the walls keep on having a probability of being occupied bigger than 0.6 and the same happens with the cells where the columns are situated and the cells of the area with moving people behind the initial robot position. There are other areas where the occupancy probability is bigger than 0.6, some with erroneous measures. The forgetting model has not deleted that information yet because it has not been applied enough times.

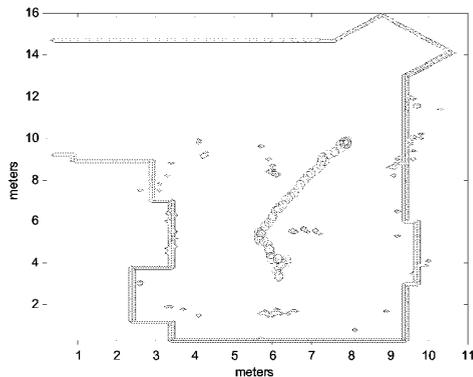


Figure 10. Occupancy probability map

The second example is presented for illustrating how the map DBN with the forgetting model behaves in case of having a moving object around the robot. In this example the robot is also asked to go from its initial position to a final position in front of it. Before the robot starts moving the fusion system detects a new object (the probability of a cell to be occupied is suddenly over the occupancy threshold) and so the robot is stopped to recalculate a new path to avoid the obstacle. The object will pass in front of the robot and all the cells it goes through will increment its occupancy probability due to the information provided by the sensors. As there is not information about the not detection of objects the probability of occupancy of those cells will be over the threshold when not forgetting is applied. The use of the DBN with the forgetting model will make the probability of those cells be decremented and so after some time nothing will exist in front of the robot which will start moving. The results of the experiment can be seen in figure 11. At the top the final map obtained from the map building fusion system is presented. The robot (yellow) remains all the time in the same place and the object (white circle) moves from the right of the robot to its left. Cells in red are considered occupied (because its occupancy probability is higher than the threshold) and cells in light blue considered free. At the bottom the states of the map around the robot (built with information of the map obtained with the DBN) in three different moments can be seen. In this submap, implemented over a quadtree, red and green squares are occupied areas while light blue squares represent empty areas. In the first submap (at the left) the object is still at the right of the robot, in the second in front of it, and in the last as its left. The occupancy probabilities of the cells around the robot have been changed initially due to the sensor information and later due to the forgetting model.

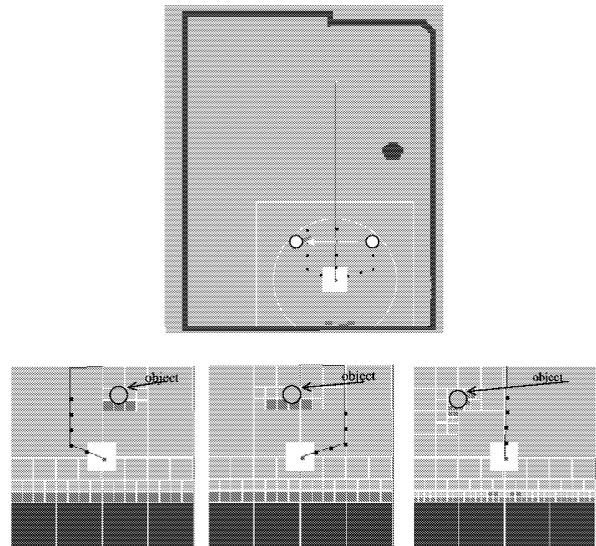


Figure 11. Occupancy map with a moving object

5 Conclusions

A unified fusion system for estimating the robot location and the state of its environment has been presented. The system is considered unified because all of it has been implemented with DBN with the purpose of having a formalized way of capturing the dependencies between the two subsystems. In the current stage of the research, our attention is more focused in exploiting the sensorial information about the robot location and map subsystems with DBN than in analyzing the dependencies among them. So, the subsystems present in the paper are independent DBN. In a second stage we will joint the two subsystems, adding arcs (and nodes if necessary), so the dependencies will be captured by the connected DBN. The structure of each DBN can be incremented for modeling more complex scenarios.

The DBN implemented for the location fusion subsystem is equivalent to the distributed information filter and can be used in a system with asynchronous sensors. The arrival of delayed information is considered in the implementation of the DBN. Evidence that arrives late to the DBN can be introduced and propagated if the delay is small enough to make the data arrive inside the window around the current time slice.

The DBN implemented for the subsystem in charge of estimating the state of the robot environment builds a map that stores the probability of each cell to be occupied or empty. The network can be used for sensors that detect presence and absence of objects. In case of not providing information about the absence of objects, the DBN also models the existence of forgetting factors.

Both DBN are generic and can be used in robots with different sets of sensors after configuring several parameters. For the DBN for robot location it is necessary to specify the state and sensor model and for the DBN for building the map the sensor and forgetting models need to be provided.

References

[1] R.C. Luo & M. G. Kay. *Multisensor Integration and Fusion for Intelligent Machines and Systems*. Ablex Publishing, 1995.

[2] M.A. Abidi, R.C. Gonzalez. *The use of Multisensor Data for Robotic Application*. IEEE Transactions on Robotics and Automation, Vol. 6, No 2, 1992.

[3] W. Burger and B. Bhanu. *Qualitative motion understanding*. Kluwer Academic Publishers. Massachusetts 1992.

[4] S. Thrun, W. Burgard, D. Fox. *A probabilistic approach for concurrent mapping and localization for mobile robots*. Machine Learning, 31:29-53, 1998.

[5] *Learning in Graphical Models*. Ed. by Michael I. Jordan. The MIT Press, Cambridge Massachusetts 1999.

[6] Finn V. Jensen. *An Introduction to Bayesian Networks*. Springer-Verlag New York 1996

[7] P. Ibaruengoytia, L.E. Sucar, S. Vadera. *A Probabilistic Model for Sensor Validation*. Proc. 12th Conference on Uncertainty in Artificial Intelligence, Portland, Morgan-Kaufmann, San Mateo, CA. 1996

[8] J.M. Regh, K.P. Murphy, P.W. Fieguth. *Vision-Based Speaker Detection Using Bayesian Networks*. Computer Vision and Pattern Recognition (CVPR99), Ft. Collins, CO, June 1999

[9] J. Sherrah, S. Gong. *Tracking Discontinuous Motion using Bayesian Inference*. Proc. of the 6th European Conference on Computer Vision. Dublin. 2000

[10] J.A. López-Orozco, J.M de la Cruz, J. Sanz , J. Flores. *Multisensor Fusion Environment Measures Using Bayesian Networks*. Proc. of the Int. Conf on Multisource-Multisensor Information Fusion. Las Vegas, USA. 1998.

[11] K.P. Murphy. *Bayesian Map Learning in Dynamic Environments*. NIPS 99. Neural Info. Proc. Systems. 1999

[12] AE. Nicholson, JM. Brady. *Dynamic Belief Networks for Discrete Monitoring*. IEEE Trans. on System, Man and Cybernetics. Vol 24, No 11, Noviembre 1994.

[13] K.P. Murpy. *Filtering and Smoothing in Linear Dynamical Systems using the Junction Tree Algorithm*. Technical Report, CS-Berkeley

[14] J.A. Lopez-Orozco, J.M. de la Cruz, E. Besada, P. Ruiperez. *An Asynchronous, Robust and Distributed Multisensor Fusion System for Mobile Robots*. The International Journal of Robotics Research. Vol 19, No. 10. October 2000.

[15] A. G.O. Mutambara. *Decentralized Estimation and Control for Multisensor Systems*. CRC Press 1998.

[16] J.A. López-Orozco, J.M. de la Cruz, E. Domínguez, E. Besada, O.R. Polo. *An Open Sensing Architecture to Autonomous Mobile Robots*. Proc. Of the IEEE Int. Symp. on Computational Intelligence in Robotics and Automation (CIRA). ISCI/CIRA/ISAS Joint Conference. Gaithersburg, MD 1998.