BUILDING MAPS BASED ON A LEARNED CLASSIFICATION OF ULTRASONIC RANGE DATA

A. KURZ

Darmstadt University of Technology, Department of Control Theory and Robotics,
Landgraf-Georg-Strasse 4, D6100 Darmstadt, Germany

Abstract. This paper introduces an approach for learning environmental maps based on ultrasonic range data. A neural network concept (self-organizing feature map) is used to learn a classification of the range data which makes it possible to discern situations. As a consequence the free-space is partitioned into situation areas which are defined as regions wherein a specific situation can be recognized. Using dead-reckoning such situation areas can be attached to graph nodes generating a map of the free-space in the form of a graph representation. In this context it is discussed how the dead-reckoning drift can be compensated.

Key Words. Classification; data reduction; learning systems; navigation; neural nets; pattern recognition; ultrasonic transducers; vehicles

1. INTRODUCTION

Many proposed approaches to the problem of path-planning for autonomous mobile robots presuppose a map which shows idealized shapes of obstacles (*geometrical maps*). When such a map of the environment is not given a priori, the autonomous vehicle must build it by itself using it's external sensors. Most of the researchers try in a straightforward way to develop methods for learning such *geometrical maps* having in mind the existing methods for path-planning (e.g. Crowley, 1989; Leonard and Cox, 1990). Work has been done in this field using different external sensor data like ultrasonic, infrared and/or laser range data. Experiments with computer-vision have also been performed.

However, it is questionable whether the high accuracy of geometrical maps is really necessary for solving the path-planning problem. A major class of path-planning methods show that this is not the case (e.g. Kampmann and Schmidt, 1989; Liu and Arimoto, 1991): these methods entail the transformation of a geometrical map into a graph to get a coarse representation of the free-space between the obstacles. The nodes of such graphs correspond to areas of free-space and the edges to possible transitions from one area to another. Using a graph-representation path-planning can be performed effectively by graph-searching techniques. A main aspect of the approach discussed in this paper is to build a graph-representation of the free-space directly from sensor data thereby avoiding the expensive transformation of the geometrical map.

1.1. *Basic Approach*

The main idea is to use an adaptive classification algorithm to process the external sensoric data, generating condensed information which can be used directly for learning a graph-representation. This works as follows: given a mobile robot which is equipped with external sensors and which is at a certain position in the environment, then the sensors yield specific data which depend on the actual position. When the robot moves the sensor data change and the size of these changes is correlated to the extent of the move, i.e., gener-

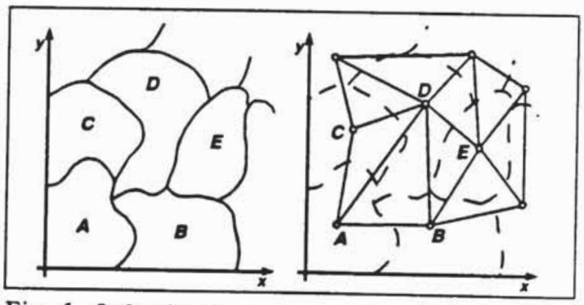

Fig. 1 Left: situation areas. Right: situation map.

ally, at similar positions the sensor data are similar too. Therefore – when similar sensor data are grouped in classes by a suitable classification algorithm – it must be possible to attach sensor data classes to specific local areas in the environment wherein the measured sensor data belong to the same class (see left side in fig. 1). Such areas can be called ***situation areas*** and the attached classes of sensor data ***situations***. Using this terminology one can say that the classification allows a recognition of situations when the robot

Original publication in IFAC Proceedings Volumes (now IFAC-Papers online), https://doi.org/10.1016/S1474-6670(17)49298-6

is in the corresponding situation areas. Based on this classification of sensor data a ***situation map*** can be built up in the form of a graph storing the situation areas as nodes and the transitions from one situation area to another as edges (fig. 1).

In the following this approach is discussed as applied to a mobile robot equipped with a ring of 24 ultrasonic range detectors as external sensors and with goniometers for estimating its position via dead-reckoning: It is shown that for recognizing situations a self-organizing feature map (an artificial neural network concept) can be used as a learning classifier (ch. 2). Further, the problem of building a map during the exploration of the environment is treated. Especially, the possibility to compensate the drift of dead-reckoning using situation maps is discussed (ch. 3).

2. CLASSIFICATION

Kohonen's self-organizing feature map (Kohonen, 1988) has been used to classify the ultrasonic sensor data. This artificial neural network concept is able to yield a vector-quantization adapted to the set of the incoming sensor data. The adaptation results in a partitioning of the training data set into classes which comprise approximately the same amount of similar members. Thereby, the Euclidean distance is used as a measure for similarity. Each neuron of the network represents a class: it responds when the signal applied to the network input (in the actual case a vector of ultrasonic range data) is a member of the class it corresponds to. Therefore, the number of possible classes is fixed due to the amount of available neurons in the network. This means in the actual context that the number of discernible situations is a (not critical) parameter which must be chosen a priori.

Before the ultrasonic range data are classified, they are preprocessed. Since situations are local entities, the measured distances are cut when they are greater then a certain threshold (in the performed experiments about 1.5 to 2.0 metres). In a second step a *virtual rotation* of the sensor ring is performed. This operation means a cyclical shifting of the 24-dimensional ultrasonic data vector which is equivalent to a rotation of the sensor ring relative to the mobile robot base. Virtual rotation is used in two different ways to make the classification independent from the robot orientation:

1. Rotation into a *reference orientation* (RO): When the orientation of the robot in the environment is known, the effects caused by the robot rotation can be eliminated turning the sensor ring virtually into a reference orientation.

2. Rotation into the *most-occupied-orientation* (MOO): For each sensor data vector a specific orientation can be calculated where the environment looks "most occupied" by obstacles (MOO), i.e., wherein the sensors measured the shortest distances. Turning the sensor ring virtually into the MOO the classification becomes also independent of the robot orientation. This method allows a variation using not all the sensors for the determination of situations, but only a certain number of sensors which lie symmetrical to the MOO. This has a positive influence making the perception of situations more local.

2.1. *Classification Results*

In an experiment a quadratic self-organizing feature map (10 x 10 neurons) has been trained with 10000 samples of ultrasonic data vectors, recorded while the vehicle moved on an arbitrarily chosen path through a laboratory environment (sample rate: 2.5/sec, velocity: 10 cm/sec).

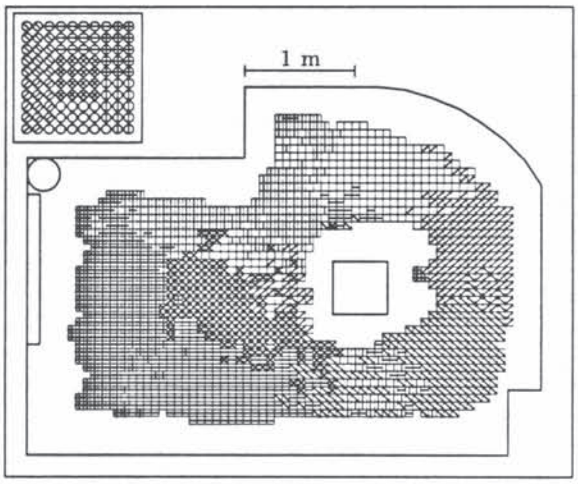

Fig. 2 RO-classification.

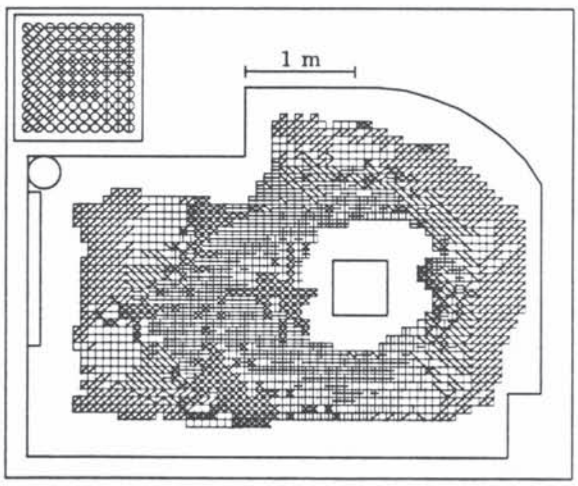

Fig. 3 MOO-classification.

The learned classification with turning the sensor ring virtually into a RO using internal sensor data (RO-classification) is visualized in fig. 2. To generate the map of situation areas shown, the environmental surface was divided into little squares and each square marked with the symbol of that neuron (situation) which responded when the mobile robot had been in the attached position. The square also shown in fig. 2 represents the Kohonen-map schematically. Several adjacent neurons have been marked with the same symbol what means that several classes have been put together and therefore only seven different situations have been distinguished. This is possible due to another feature of Kohonen-maps: adjacent neurons (situations) in the network correspond to adjacent situation areas in the environment.

The result of learning a classification using the MOO-calculation (MOO-classification) is shown in (fig. 3): Comparing it with the RO-classification, the situations correspond e.g. to things like corners, which can be oriented arbitrarily in the environment (areas near corners are marked equally which means that the same situation has been recognized there).

3. MAP-BUILDING

3.1. *Generating a Graph-Representation*

When the training of the self-organizing feature map (for the situation-classification) has been finished the mobile robot starts a second exploratory run for map-building. In every moment, when the classification indicates a change of the situation, it has crossed a situation area. Then - when not already existing - a new node and a new edge are added to the graph-representation of the free-space (the map) representing the just traversed actual situation area and the transition from the last to the actual situation area. In this way a *topological* situation map is generated since only relations of neighbourhood and no metrical data are stored. As already mentioned in the introduction, such a topological graph-representation is sufficient for path-planning, but there are reasons for the necessity to add metrical information to the graph:

— Since the environment may look similar to the robot in different areas, situation areas are existing which correspond to the same situation. They can only be distinguished when distances between situation areas are known.

— For following a planned path it must be known how to get from one situation area to another. Therefore, at least rough orientations and distances must be available.

To meet these requirements to each graph node some specific position in the environment is attached. This means that representative positions for situation areas have to be found. Figure 4 shows how this problem is solved by averaging all those positions of the robot which belong to pieces of the trace which cross the same situation area. The so learned map of the same environment shown in fig. 2 is visualized in fig. 5. It is the result after 30 min. of exploration (velocity: 10 cm/sec) using the trained feature map of fig. 2. For reasons of better quality a slightly enhanced position correction has been applied which allows big situation areas to be representated by several graph nodes.

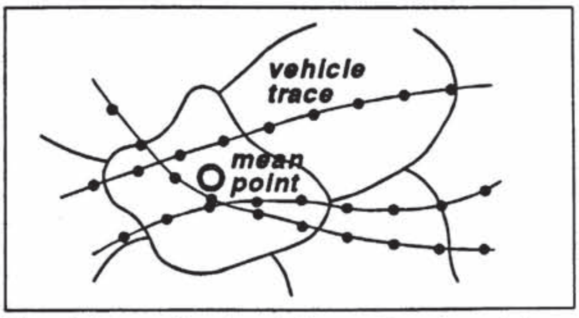

Fig. 4 Determination of the graph node positions.

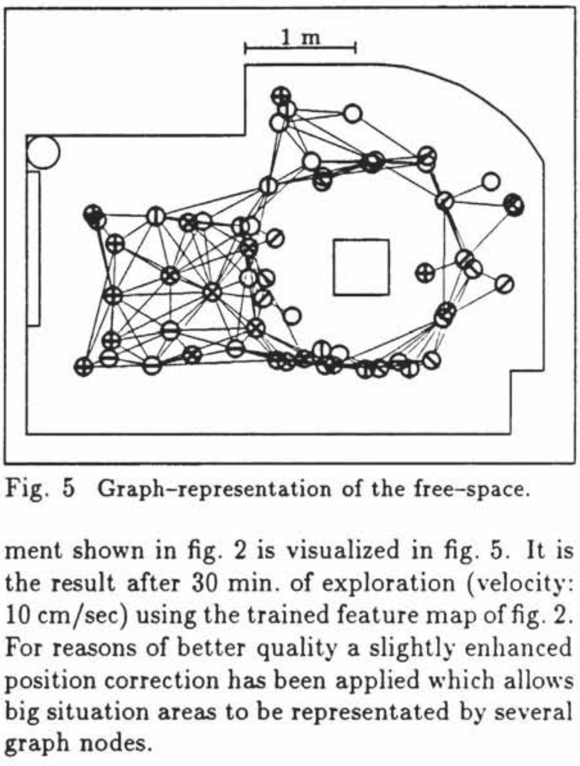

Fig. 5 Graph-representation of the free-space.

3.2. *Position Estimation*

To build up the graph the position of the mobile robot must be known because it is the basic requirement for the calculation of the graph node positions. Experiments confirmed that a relative coarse estimate suffice, when the error is bias-free: Since the node positions are the result of an averaging process deviations from the exact position are smoothed. The inaccuracies which are not filtered in this way can be accepted since node positions are coarse statements. For the map shown in fig. 5 a position estimate yielded by dead-reckoning (deviation about 1% of the covered distance) has been used, which has been corrected approximately every 30 metres (the robot corrected the position estimate by itself using its ultrasonic sensors in a special procedure when it reached a designated corner with known position).

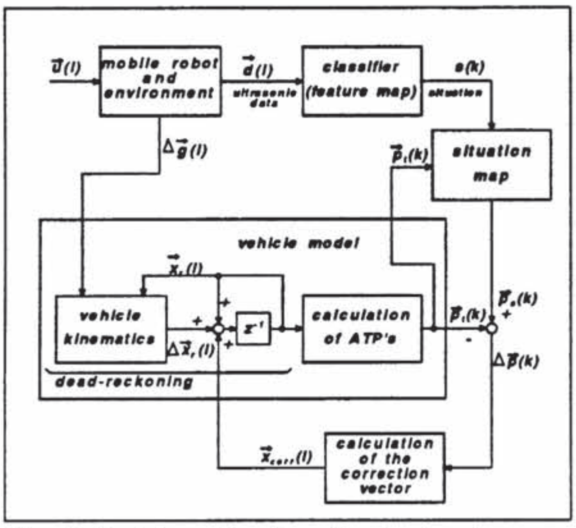

Fig. 6 Position estimation using the situation map.

It is also possible (but a much harder problem) to use the situation map itself for the correction of the dead-reckoning error: the positions of situation areas which are registered there can be compared with the dead-reckoning position and the observed difference can be employed for drift compensation. The diagram depicted in fig. 6 shows this idea more precisely:

— While the mobile robot moves through the environment driven by the motor control values $\vec{u}(l)$ its goniometric sensors measure differential angles at the motor axes, $\Delta\vec{g}(l)$, and its ultrasonic sensors yield distances to obstacles, $\vec{d}(l)$ at every time step l.

— The ultrasonic sensor data $\vec{d}(l)$ are analysed by the classifier. At each time step k, when the recognized situation changes, the robot has crossed a border between two situation areas. Then, the situation $s(k)$ is recorded which corresponds to the situation area just passed through (see fig. 7 which illustrates the relationship of the time steps l and k).

— The goniometric data $\Delta\vec{g}(l)$ are used for performing dead-reckoning to estimate the internal state $\vec{x}_r(l)$ of the robot. This state is determined by three values: two coordinates of the robot position, $x(l)$ and $y(l)$, and the robot orientation $\alpha(l)$.

— At every time step k the position estimates which belong to the traversed situation area are averaged calculating ATP's (*averaged trace points*) $\vec{p}_t(k) = (x_t(k), y_t(k))^T$ (see fig. 7).

— Using the ATP $\vec{p}_t(k)$ and the attached situation $s(k)$ the crossed situation area can be determined in the situation map. It corresponds to that graph node which is attached to the recognized situation $s(k)$ and possesses the graph node position $\vec{p}_s(k)$ which is nearest to the ATP.

— The difference $\Delta\vec{p}(k)$ between the ATP $\vec{p}_t(k)$ and the position of the just crossed situation area $\vec{p}_s(k)$ is used to calculate a correction vector for the estimate of the internal robot state, $\vec{x}_{corr}(l)$.

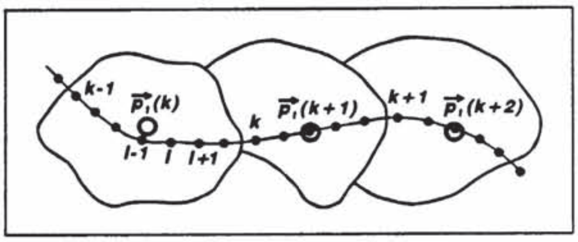

Fig. 7 Averaged trace points and time steps l, k.

The structure shown in fig. 6 can be interpreted as an observer for estimating internal system states, a well-known system-theoretical concept: The mobile robot, the classifier and the situation map are the process with the internal state $\vec{x}_r$ of the robot, the dead-reckoning and the calculation of ATP's correspond to the process model of the observer.

The correction vector comprises in correspondence to the robot state three components: $\vec{x}_{corr} = (x_{corr}, y_{corr}, \alpha_{corr})^T$. The first and the second can be combined to the vector $\vec{p}_{corr} = (x_{corr}, y_{corr})^T$ which is a correction vector for the robot position. It can be calculated in a straightforward way from the observed difference between the ATP and the graph node position of the traversed situation area:

$$\vec{p}_{corr}(k) = k_p \Delta\vec{p}(k) = k_p(\vec{p}_s(k) - \vec{p}_t(k)) \quad (1)$$

The correction factor k_p is a parameter which has to be specified. Figure 8 visualizes the correc-

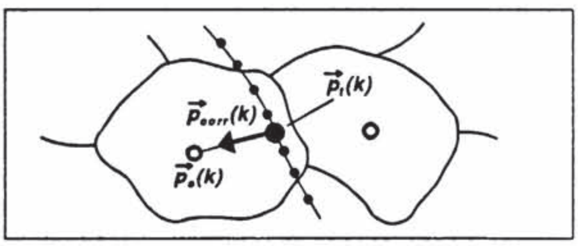

Fig. 8 Position correction.

tion procedure. When the trace of the mobile robot in the situation map (the sequence of the positions of visited nodes) is interpreted as the sum of the real trace in the environment corrupted by additional (rather strong) noise, k_p can be considered as a constant noise filter factor. Further, the structure in fig. 6 is then a system for compensating uncertaincies of the process model (resp. the drift of dead-reckoning) using noisy process output values (resp. positions of graph nodes in the situation map).

The calculation of the third element of the correction vector, α_{corr}, is more difficult: An estimate of the robot orientation α_r is available from dead-reckoning, but the situation map yields only estimates for positions $\vec{p}_s = (x_s, y_s)^T$. Therefore a difference between an observed and the estimated orientation ($\Delta\alpha$) can not be got directly to calculate α_{corr} in analogy to equ. 1:

$$\alpha_{corr}(k) = k_\alpha \Delta\alpha(k) \qquad (2)$$

In the following several approaches for the correction of the robot orientation are discussed. They differ in the way how values for $\Delta\alpha$ are acquired using different types of situation maps. They have been tested in experiments using situation maps which have been learned a priori – the problems which arise when position estimation and map building are performed simultaneously are discussed in the next paragraph.

Three-dimensional situation areas. A first idea was to build situation maps without turning the sensors (virtually) into a special orientation. Then situation areas in the three-dimensional state space of the robot can be defined and points of the state space can be attached to the graph nodes. However, experiments revealed that the position estimation is not stable.

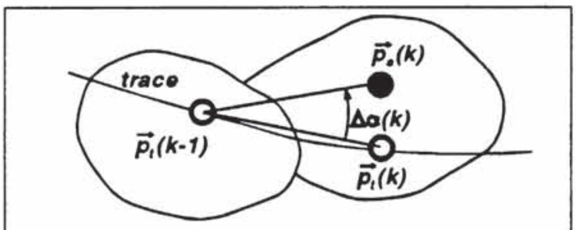

Fig. 9 Correction of the robot orientation.

Situation Maps Using RO-Classification. When the mobile robot is driven by two motors, it is possible to infer the robot position from the trace, and the tangential orientation of the trace curve can be taken as the internal state variable α. Then $\Delta\alpha(k)$ can be calculated as the angle formed by the two ATP's $\vec{p}_t(k)$ and $\vec{p}_t(k-1)$ and the node position $\vec{p}_s(k)$ (see fig. 9). This method works, but it is not very robust due to the fact that the reference orientation which is needed for the RO-classification must be derived from the estimated state variable α.

Situation Maps Using MOO-Classification. This does not hold when MOO-classification is performed. Then the angle for the virtual rotation of the sensor ring is calculated using the ultrasonic data exclusively and the position estimation is very reliable. Alternatively, the independency of the MOO from the estimated α_r makes it also possible to store the MOO-information in the map, and to use it for getting a $\Delta\alpha$ by calculating the difference between the MOO of the actual ultrasonic data and the MOO which is entered in the map with the corresponding graph node. Both methods are useful and robust: the adjustment of the parameters is not critical and the inaccuracy of the dead-reckoning can be relatively great (ca. 1.8 degrees per turn of the mobile robot).

3.3. *The Map-Building Problem*

The position estimation method works well if a situation map already exists. Further, it is also no problem to build up the graph-representation and to determine the positions of its nodes when the position of the robot is known exactly. However, is it also possible to build the situation map and to use it for position estimation at the same time? This is the general problem of learning maps by a mobile robot which is equipped with goniometers for measuring its position (dead-reckoning): The position estimate has to be corrected using a map which only can be learned using this corrected position.

The experiences from experiments performed with a real mobile robot in laboratory environments can be summarized in the following statements:

— It is no problem to build stable situation maps when the orientation of the mobile robot is known and the position drift has to be compensated only. Then, both RO-classification and MOO-classification can be used.

— If the drift of the robot orientation must be corrected too, it is not possible to build stable situation maps using RO-classification. With MOO-classification, maps can be built up but only when the drift of the orientation is relatively small and when the exploration trips into unknown areas are not too long.

In fig. 10 an example of a stable graph-representation is shown which has been learned in a greater environment during about 30 minutes of exploration using MOO-classification. The little bars at the graph-nodes represent the MOO-information stored in the situation map.

Above all, problems arise when the mobile robot explores a fully unknown area. Then, it must rely on the accuracy of dead-reckoning and of the last position and orientation estimate which has been calculated when it has left the known area. As

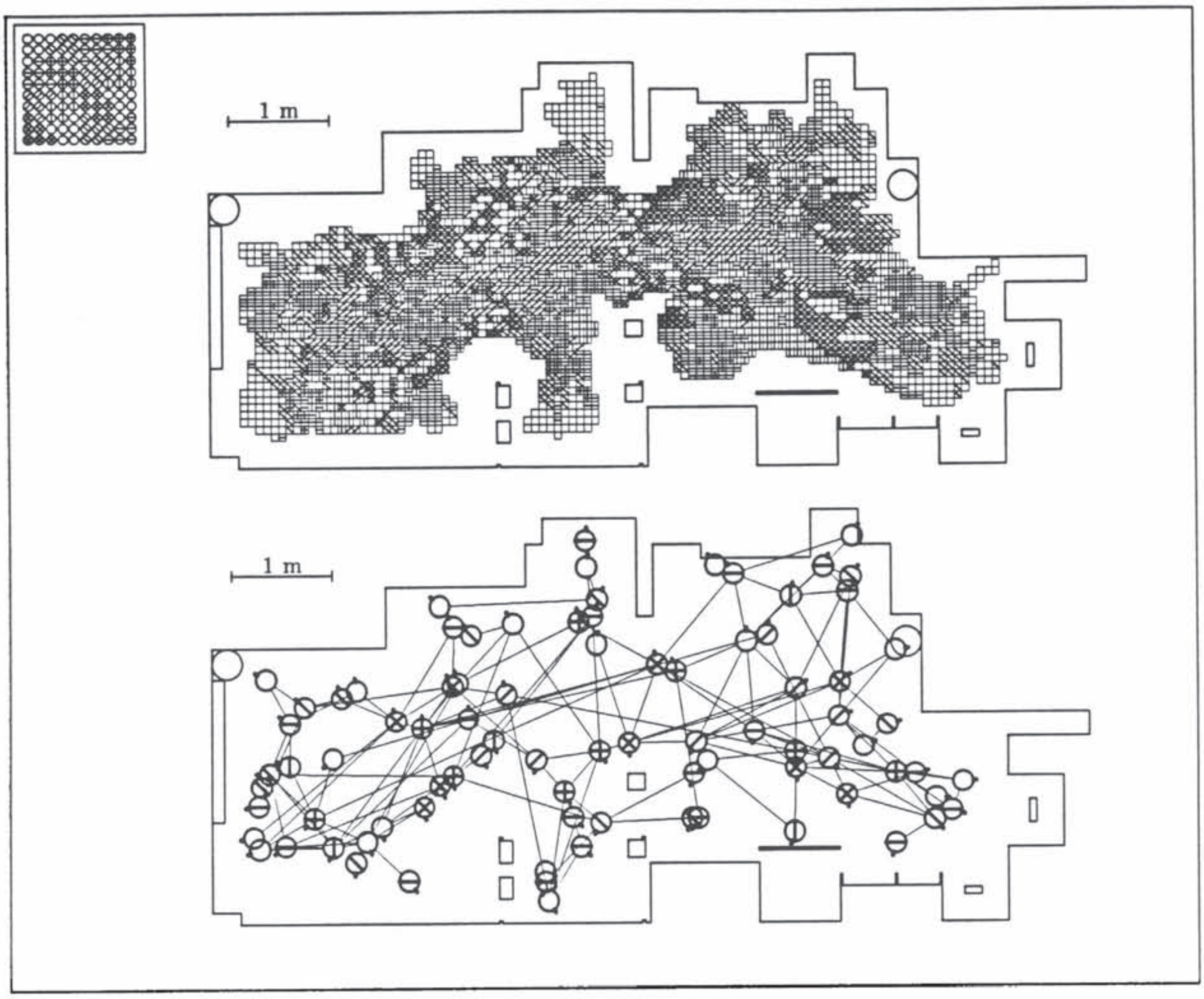

Fig. 10 Map-building with unknown robot orientation using MOO-classification.

a result, differences between exact and estimated position influence the determination of the graph node positions and are documented in the map. The error in the orientation estimate especially has grave consequences since it leads to position errors which grow with the distance proportionally. Therefore, the situation map contains inaccuracies and inconsistencies, which make the position estimation very difficult.

4. SUMMARY

In this paper a method has been introduced for generating graph-representations of a mobile robots world directly through evaluation of ultrasonic range data: in the first step a self-organizing feature map is trained to learn a classification of the ultrasonic data which allows the recognition of situations. In the second step a graph is built up attaching to situation areas specific positions which are calculated during the exploration of the environment. Especially it has been discussed under which conditions situation maps can be used to compensate the drift of a dead-reckoning position estimate.

5. REFERENCES

Crowley, J.L. (1989). World modelling and position estimation for a mobile robot using ultrasonic ranging. *Proc. Int. Conf. Robotics and Automation*, 674-680.

Kampmann, P. and G.K. Schmidt (1989). Multilevel motion planning for mobile robots based on a topologically structured world model. *Conf. Proc. Intelligent Autonomous Systems*, Amsterdam, The Netherlands.

Kohonen, T. (1988). *Self-Organization and Associative Memory.* Springer, Berlin, London, second edition.

Leonard, J. and I.J. Cox (1990). Dynamic map building for an autonomous mobile robot. *IEEE Int. Workshop on Intelligent Robots and Systems.*

Liu, Y. and S. Arimoto (1991). Proposal of tangent graph and extended tangent graph for path planning of mobile robots. *Proc. IEEE Int. Conf. on Robotics and Automation.*

Acknowledgement: this work has been supported by Deutsche Forschungsgemeinschaft (grant To-75/17-2).