Semantic Place Classification and Mapping for Autonomous Agricultural Robots

Ulrich Weiss and Peter Biber

***Abstract*— Our work focuses on semantic place classification and on navigation of outdoor agricultural robots equipped with a 3D laser sensor. We classify the environment semantically and use this information for monitoring, mapping and high level navigation planning. On a field limited classes of locations like crop rows, open field as well as the beginning and the end of rows are expected. These different classes of locations and the transitions between them can be encoded using a stochastic automaton. To calculate the probability distribution over its states and associated variables, we are using a particle filter. This allows us to easily add further sensor data and knowledge about the environment. Currently we determine the probability to be in a state by using patterns based on 3D laser sensor data and compare them with state specific patterns. Further we are using the knowledge about the approximate row length. Situations that cannot be predicted this way force the automaton to switch to an error state which stops the robot immediately. The experiments conducted using the simulation environment Gazebo and real maize field data have shown that we are able to classify the environment with rates of around 97%. If we compensate a short time delay the classification rate becomes nearly 100%.**

***Index Terms*— semantic mapping and place classification, semantic robot vision, agricultural robotics**

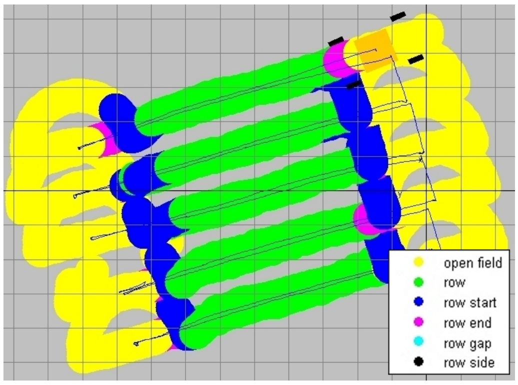
I. Introduction

In agricultural robotics and for the automatic guidance of tractors mostly RTK-GPS based approaches for localization and navigation are used. But these approaches have some limitations: the vehicles need to be taught to the field, or have to use previously recorded tracks or seeding positions, e.g., from drilling vehicles. The RTK-GPS devices are also very expensive. Besides the pose of the autonomous vehicle the environment around it is of interest. If the vehicle is able to recognize its surroundings and to classify it, it improves the localization, navigation and mission planning. If navigation is based solely on position, classification of the surroundings can also be used to monitor the execution of plans.

Our work focuses on the localization and navigation of such autonomous vehicles in outdoor agricultural environments. Especially the case of row cultivations is of interest. It is a part of the publicly funded project BoniRob [10], a cooperation of academic and industrial partners, with the goal to develop an autonomous mobile agricultural robot. The key objective of BoniRob is the autonomously repeating phenotyping of individual plants on plant breeder's lots at different days, to track the growth stage and the condition of the plants. For that, a robust robot platform is needed which is able to navigate autonomously on unknown fields. The robot platform developed by the project partners is shown in Figure 1(a). It has four individual steerable and drivable wheels and is able to adjust the height clearance of the chassis as well as the track width. This enables the robot to adjust to varying field conditions.

We think for the localization and navigation of such a robot and for many other agricultural robotic applications, an approach which does not only rely on detailed previous knowledge about the crop positions is more sufficient. For example, this is the case on small fields which are not seeded with huge machines equipped with high precision but expensive GPS devices. The robot should also run on previous unknown fields. Due to that, we pursue an approach which builds-up a topological semantic map of the field (see Figure 1(b)) and uses these information for navigation planning. To plot the semantic maps a high precise GPS receiver is used, but for the navigation planning and the

(a) BoniRob Platform



(b) Semantic Field Map

Fig. 1. The pictures in (a) show the BoniRob platform on the maize field corresponding to the map underneath. The pictures were taken at the DLG Field Days 2010. The diagram in (b) shows an example of a semantic map of a maize field with a row length of around 6 m. The blue line indicates the path traveled by the robot; the colored circles the semantic map. The positions of these circles correspond to the position of the spot in front of the robot sensed by the 3D laser sensor.

Ulrich Weiss and Peter Biber are with the Corporate Sector Research and Advance Engineering, Robert Bosch GmbH, Schwieberdingen, Germany {ulrich.weiss,peter.biber}@de.bosch.com

monitoring only semantic information are required. Currently we localize the robot according to the kind of environment it is in, but the semantic information could also be used to support the positioning.

On a field we are expecting a limited number classes of locations like crop rows, open field as well as the beginning and the end of rows. These locations can be encoded using a probabilistic state automaton. The key idea of this paper is to determine the *semantic state* of the robot based on such an automaton which describes the transitions between the different classes of locations. We calculate the probability to be in a specific state by using a particle filter. This approach enables us to easily add further sensor data and knowledge about the environment. Each particle contains a state-id and the probability to be in this state as well as additional information like the distance traveled since the last state transition. The transition probabilities between the states were experimentally identified and depend on the movement of the robot.

To classify the environment, we determine simple patterns based on laser sensor data describing the environment and compare them with predefined *state patterns*. All state patterns together form the state pattern database, which can be easily replaced by other databases. If available, we are additionally using the approximate row length and the traveled distance as an additional information for weighting the particles. To encode unpredictable situations, the automaton uses an error state which stops the robot immediately. This is a step towards a more robust approach against malfunctions and a more safe system. All situations that cannot be explained by our semantic classification end to the error state. Therefore the robot only moves in situations that have been foreseen explicitly by the application developer. Autonomous navigation in unclear states cannot be allowed for safety-critical applications like ours.

This paper is organized as follows. In section II related work dealing with semantic place classification, mapping and localization are discussed. Our semantic place classification and mapping algorithm is presented in section III. Section IV shows the experimental results. The paper closes with conclusions and future work in section V.

II. Related Work

Semantic mapping and classification of places is becoming a big issue in mobile robotics and is of interest in a wide variety of robotic applications. In the last years, several authors considered the problem of adding semantic information to places. For example Buschka et al. [1] have used a virtual sensor to identify rooms using range data. Torrabla et al. [11] presented a context based vision system for place and object recognition. They aim to identify familiar locations, to categorize new environments and use that information to provide contextual priors for object recognition. Therefore they used low-dimensional global image representation and trained their system with over 60 in- and outdoor locations. Another approach for location estimation was presented by Oore et. al. [8]. They trained a neural network that maps locations into likelihood distributions across possible ultrasonic readings. Kuipers et al. [4] used bootstrap learning for place recognition which does not need any prior knowledge of its sensors, effectors or the environment.

Besides using semantic information for object or place recognition, these information can also be utilized to build and interpret maps. Nüchter et al. [7] proposed a 3D mapping algorithm for mobile robots using a high resolution 3D laser sensor and coarse scene features like walls and floors in buildings. More delicate objects are detected by trained classifiers. Another work using a 2D laser sensor and a mobile robot to build semantic annotated maps was presented in [2]. They used and compared a hand-crafted and a learning classifier to generate indoor grid maps and enhance these maps with semantic information.

Approaches to improve the localization of a mobile robot in a domestic environment using semantic information were proposed by Rottmann [9] and Mozos [5]. They used supervised learning and boosting for semantically labeling different locations. For that they trained a classifier using features extracted from vision and laser range data. Afterwards they applied a Hidden Markov Model (HMM) to increase the robustness of the classification.

In contrast to the above described approaches, our algorithm is using a particle filter, which enables us to easily add further information describing the state as well as additional sensor data. To classify the states, we distinguish very easy and fast to calculate patterns based on 3D laser sensor data and compare them with an exchangeable pattern database. Due to that, we can easily switch, e.g., between the detection of one or two rows. We also defined an error state to deal with unpredictable and unforeseen situations, to make the approach more robust against malfunctions.

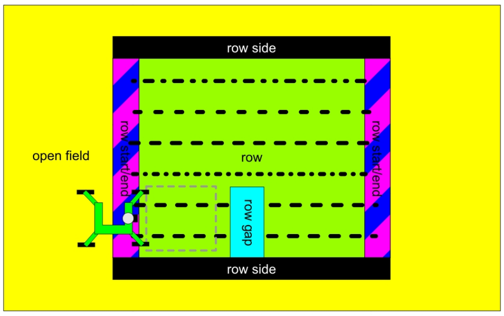
III. Semantic Place Classification and Mapping

A. Field Partitioning

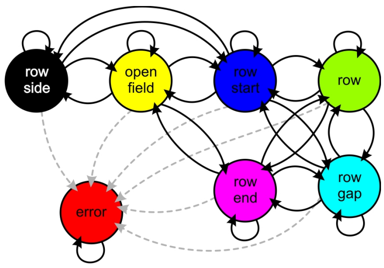
By considering an agricultural field one sees that it can be partitioned into a limited number of semantic locations. Due to that, we partitioned the field into six different classes: open field, crop rows, row gaps as well as the begin, the end and the side of the rows (see Figure 2(a)). We define these locations plus an error state as the state set

$$S = \{\textit{open field},\ \textit{row},\ \textit{row start},\ \textit{row end},\\ \textit{row gap},\ \textit{row side},\ \textit{error}\}.$$

The *row gap* state indicates missing plants in rows and is closely related to the *row* class. If the field is not seeded in rows, the *row* and *row gap* class can be replaced by other location classes. Because our robot can adjust its track width it is able to drive over one or over two rows. Due of that, the *row* and the *row gap* locations can show different number of rows. The number and classes of locations depend on the application the developer does consider. A possible extension of the field partitioning is to break down *open field* into *headland* and *off-field*, but this is not required for our application.

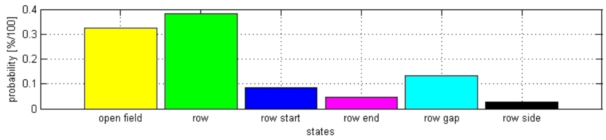


(a) Semantic Field Partition

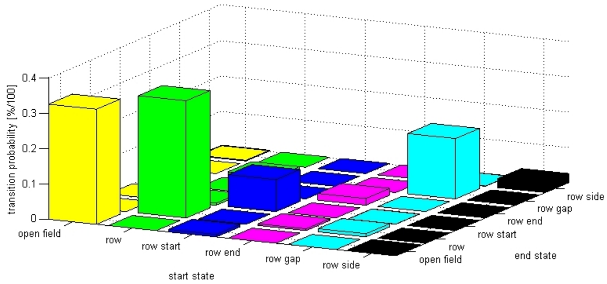


(b) Stochastic Field Automaton A

Fig. 2. The semantic field partitioning in (a) shows our proposed segmentation of a typical row cultivation. The dashed square in front of the robot displays the spot sensed by the 3D laser sensor. (b) shows the automaton A which describes the field partitioning and transitions between the field partitions. The additional *error* state is used for unpredictable situations, e.g., if an obstacle is sensed. In each state there is a certain probability to switch into this *error* state.



(a) Average State Probability Distribution



(b) State Transition Probabilities for A_{move}

Fig. 3. The diagrams (a) and (b) show the experimentally determined state and transition probability distributions. A_{stay} is not displayed because the probabilities to stay in the same state are 100%. The probabilities for the *error* state are also not displayed because they cannot be determined by experiment.

In our case we are aiming to navigate autonomously over row cultivations, e.g., maize. While the robot is traversing the rows it should follow them until it reaches the end of the rows followed by a turn on the open field. Afterwards it searches for the begin of a row and starts the row following again. Using the semantic place classification the robot is able to derive navigation instructions to execute this navigation plan. Besides for generating navigation instructions the place classification can be used to monitor the navigation. E.g., a turn maneuver should be only carry out, if the robot is on the open field.

B. Semantic Place Classification

As Figure 2(a) shows only a limited number of transitions between the different states are possible. These states and the possible transitions between them are encoded using the probabilistic state automaton A (see Figure 2(b)). A consists only of transitions between states which are possible according to the semantic field partitioning and reasonable for the navigation. For example, the robot should not drive from *row side* into *row*. To deal with such circumstances and other not predictable situations we implemented an *error* state. If this state is reached the robot should stop immediately until the situation is cleared.

The state probabilities and transition probabilities between the states were determined by using a large number of example data recorded using a simulation environment. For that, we vary field parameters like the row length, the plant height, the gap probability and the number of rows. Two different state automata A_{stay} and A_{move}, similar to the approach in [9], were determined. A_{stay} indicates the transition probabilities if the robot stays at the same place and A_{move} if the robot moves. Figure 3 shows the average state probability distribution and the transition probabilities for A_{move}. For A_{stay} the probability to stay in the same semantic location is set to 100%. To all of the states of both automata a certain probability to switch into the *error* state is added. We defined this probability uniformly distributed over the states and used 10% in our experiments. By changing these percentage we can influence the probability of error detection.

The probability distribution over the states is calculated using a particle filter with particle injection. Each particle in the filter contains a state-id $s_{id} \in S$ and an importance factor $w \in [0,1]$, but can also contain additional information, like the distance traveled since the last state switch of the particle, which are used in the observation model. For calculating the probabilities, the standard particle filter equation

$$bel(x_t) = \eta p(z_t, x_t) \int p(x_t | x_{t-1}, u_{t-1}) bel(x_{t-1}) dx_{t-1}$$

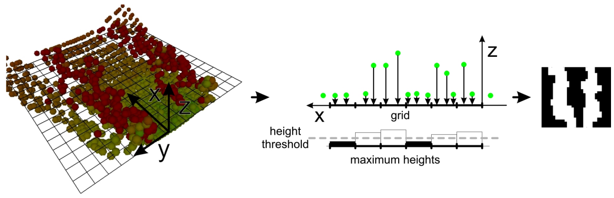
is used. With x, the importance factor distribution, z, the measurements, and u, the robot motion. u is set to *stay* or *move* depending on the robot movement. Depending on u, A_{stay} or A_{move} is chosen to calculate the motion model $p(x_t | x_{t-1}, u_{t-1})$. In the observation model $p(z_t, x_t)$, the probabilities of the sensor measurements are multiplied. For the start distribution of the filter and for the injection of new particles the state probability distribution in Figure 3(a) is used. The particles change their s_{id} in the sampling step of the particle filter each time a new velocity observation is received. These observations should be time-equidistant. For this state switch the transition probabilities of A_{stay} respectively A_{move} are used. In the measurement step the particles weights are recalculated. These weights are determined using the probability of a measurement according to the current s_{id}. Currently we are using a 3D laser sensor and the approximate row length for the particle weighting.

This probabilistic approach enables us to easily add additional information by using further sensors in the measuring step, like vision or GPS. If we would only use the probability of the states a HMM approach is sufficient, but we would like to be able to use further multi modal distributed information. In such a case a particle filter gives us the freedom to do so. If required, additional variables are added, like the GPS position or the traveled distance since the last state switch. But many more are imaginable. The number of particles in the filter depended on the number of additional variables. In our experiments we used 200 particles.

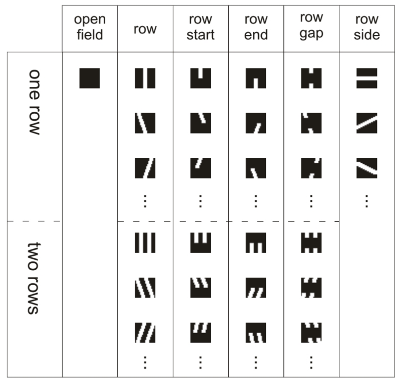
C. Place Classification using a 3D Laser Sensor

In this section the pattern detection and environment classification using a 3D laser sensor are described. Our robot is equipped with a low resolution 3D laser sensor with 29x59 pixels. It is mounted on the front of the robot and is pointing towards the ground with a pitch angle of around 45°. By using this sensor we determine simple environment describing patterns. For calculating these patterns, we detect the ground plane using the RANSAC algorithm. The plane detection algorithm was already described in the previous publication [12]. Afterwards we transform the point cloud into the plane coordinate system and place a squared grid of 20x20 cells and a side length of 2.0 m on this plane. For each grid cell the maximum height of the points above the ground plane is determined and a binary image using a fixed height threshold of 0.075 m is calculated (see Figure 4(a)). Using the measured plant height, this threshold could be adjusted, but our experiments had shown that this fixed threshold works fine for all of our tested cases with plant heights between 0.15 and 1.5 m.

To classify the state we compare the binary images with a pattern database. In this database for each state — except the *error* state — a number of patterns are defined. Figure 4(b) shows an example pattern database. These patterns have to be aligned to the application. E.g., in maize row cultivations the row displacement is standardized to 0.75 m. This enables the robot to detect if the field looks like as expected. A big advantage of this approach is, that we are able to easily exchange or create new databases, for example if we would like to drive over two instead of over one row. For each of the patterns in the database we automatically calculate some additional variants by translating and rotating them. These whole database is correlated with the currently detected binary image using sum of squared differences. Finally, the maximum of the correlation for each state is determined. These maxima are used as the probabilities in the measurement step. For the *error* state a constant correlation value of 0.5 is used. Figure 5 shows the correlation values of an example run and the resulting state distribution of the particle filter.

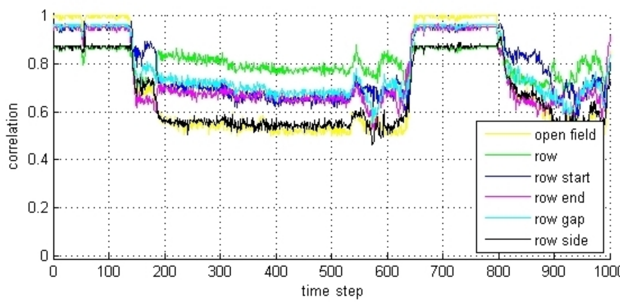


(a) Pattern Calculation

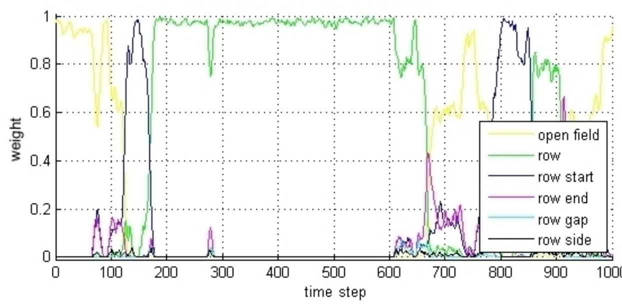


(b) Pattern Database

Fig. 4. The scheme in (a) shows the steps of the laser pattern calculation. First a grid is placed on the sensed ground. Afterwards the points are projected on this grid. Finally, we determine a binary image by comparing the maximum values of each cell with a fixed height threshold of 0.075 m. This images are correlated with a pattern database. Table (b) shows a subset of this database. For each state of S — except for the *error* state — a set of patterns is defined. For the states *row*, *row start*, *row end* and *row gap* different patterns for one and two rows are used. Depending on the application different subsets of this database can be chosen.



(a) Pattern Correlation



(b) State Probability Distribution

Fig. 5. The diagrams (a) and (b) display the pattern correlation and state probability distribution for a typical run over a row, followed by a turn on the open field and starting row following again.

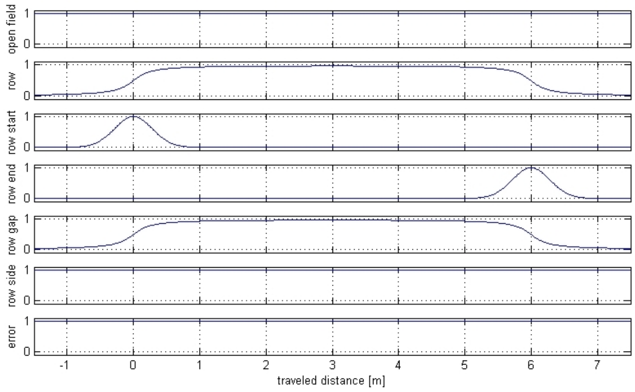


Fig. 6. Row length weighting functions for the different field locations and an estimated row length of 6 m, with a high certainty.

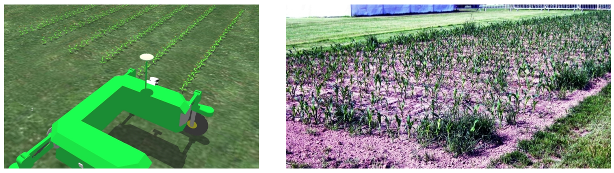
D. Particle Weighting using the Row Length

If many plants are missing in a row, such gaps can look similar to *row end* or *row start*. To improve the semantic classification we use the knowledge about the approximate length of a row and add an additional variable to the particles which cumulates the driven distance. This variable gives us a distribution over the driven distance for all particles in the same state. The particles are weighted using the driven distance and the estimated row length. Of course, this weighting is only possible, if the approximate row length is known. But the row length can be easily measured using the robot, e.g., by marking the first row end by the operator.

For each particle the weights using the row length are calculated as follows. If the s_{id} of the particle equals to *open field* or *row side* we set the distance variable to 0 and cumulate the driven distance in all other cases. In the measurement step the weight of the particle is multiplied with the calculated weight using the the driven distance variable and the functions shown in Figure 6. These functions depend on the row length and the uncertainty about it.

IV. Experiments

The goal of the experiments is to demonstrate that our approach provides a robust localization of an agricultural robot on a field in a semantic manner. The experiments were conducted using the BoniRob platform. For environment sensing we used the low resolution FX6 laser sensor [6], which was already introduced in the previous publication [12]. The proposed algorithm has been implemented and tested using the simulation environment Gazebo [3] and was evaluated using real field data. Figure 7 shows these testing environments. The simulation environment allows us to use different field setups and to record a large number of example data. Using this data we were able to derive the state and transition distributions. Because our use case focuses on fields of plant breeder's we used fields with a row length of around 6 m, but we tested our algorithm also on fields with longer and short rows, receiving similar results. To evaluate the results hand-crafted ground truth data for the simulation environment as well as for the real maize field were used.



(a) Simulated Field (b) Real Maize Field

Fig. 7. Testing environments

TABLE I
State Estimation Results

field type	gaps	Pattern Correlation [%]	Particle Filter [%]	Particle Filter + Row Length [%]
real field	without	93	96	97
sim. field	without	94	96	97
	some	83	85	90
	many	59	62	82

The result of the classification are shown in Table I. Using the simulation environment we performed our tests driving over one and over two rows simultaneously. Also, we used different field setups with no, some and many gaps in the rows. The real field data (see Figure 7(b)) does not show any gaps, so we just got results for no gap situations. Table I shows the average results of several test runs with a few thousand laser scans. The table compares the results of the classification using pattern correlation, the particle filter with pattern correlation, and the particle filter with pattern correlation and the row length measurement. In both cases we used 200 particles.

For the simulation and real world experiments we received almost the same results. If no gaps are shown in the rows the pattern correlation yields good results. The more gaps the classification decreases, because the gaps are false classified as *row end* or *row start*. Using the particle filter the classification rate increases. If there are many gaps the additional knowledge about the row length increases the classification rate.

Because there is a short time delay the table misleads about the real classification quality. The more state switches, the more this circumstance takes effect. This gets more obvious if the gap number increase. In Figure 8 one can see these delay for an example state sequence. The offset between the ground truth and the classification data is approximately constant. These offset is varying slightly because of the low pass nature of the particle filter and subjectiveness of the ground truth data generation. Also, there is a smooth transition between two field locations. By determine the average offset and shifting the particle filter sequence the classification rate increases to nearly 100%. This is sufficient enough for navigation.

If we consider the sequence of states and filter out the states which are only detected for one or two time steps we are able to derive high level navigation commands using the semantic place classification. Figure 9 shows two resulting maps of a simulation and a real world experiment. To plot

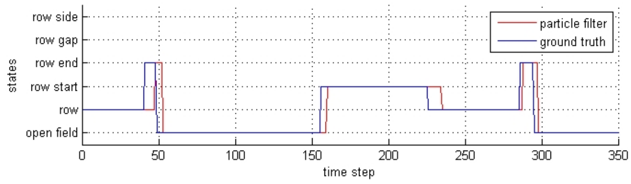
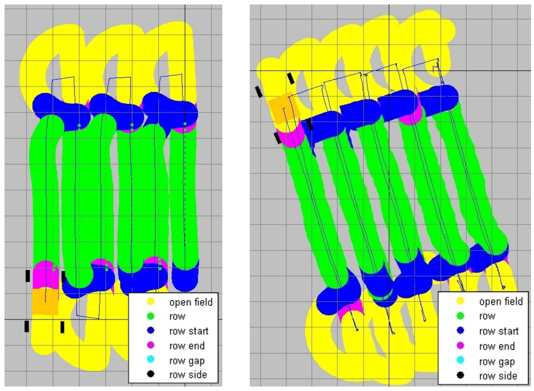


Fig. 8. The Figure shows a detected state sequence using the particle filter and the corresponding state sequence of the ground truth data. For each state switch a delay of the particle filter can be recognized.



(a) Simulated Field (b) Real Maize Field

Fig. 9. (a) shows the resulting semantic map using a simulated field with rows of 5.5 m length. The small green points display the plant positions. In this experiment the robot droves over one row. (b) shows the semantic map of a real maize field corresponding to the image in Figure 7(b) and driving over two rows.

these maps we localized the robot using a Kalman filter fusing GPS, odometry and IMU data.

We also did some tests concerning the error detection using the *error* state. This state was reached several times in situations where the robot turned in the middle of the rows or when a person stepped into the area sensed by the laser sensor. But a qualitative evaluation has not yet been conducted, but will be done.

V. Conclusions and Future Work

A. Conclusions

In this paper, we presented a novel approach to classify different places in an agricultural environment. For that, we partitioned the field into different classes of semantic locations and used a stochastic automaton describing the relations between them. Using this automaton and a particle filter we determine the probability to be in one of these semantic locations. This approach can be used to improve the localization, navigation and mission planning as well as to monitor the execution of plans. To deal with unpredictable situations, we added an error state. This error state is a step towards a greater robustness against malfunctions. We are using a 3D laser sensor to determine fast and easy to calculate patterns. These patterns are used to classify the semantic locations. We compare these patterns with a pattern database, which can be easily replaced by other databases. For example, when the robot should drive over two rows simultaneously instead of over one row other patterns for row classification are required. Our approach allows us to easily add additional sensor data or knowledge about the environment. Such knowledge could be the row length or the field borders.

Our algorithm has been implemented and tested on a real robot as well as in simulation. The experiments have shown that we are able to classify the environment with classification rates of around 97%. If we compensate a short time delay this rate becomes nearly 100%, sufficient for navigation planning.

B. Future Work

Because our work is still in progress there are many ideas and things to do. For example, we would like to compare the particle filter approach with a HMM. Currently we are using experimentally specified transition probabilities. Learning these probabilities online is a future topic, also the learning of the state pattern is of interest. Because we are using an omnidirectional robot it may improves the classification if we distinguish between more different kind of movements. Also, we are currently working on the state classification using further sensor data and additional knowledge like a vision sensor or the field borders. Further, more analyses have to be done, e.g, how reliably errors can be detected. And of course a further big issue is to integrate the semantic classification in the positioning of the robot.

References

[1] P. Buschka and A. Saffiotti. A virtual sensor for room detection. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 637–642, Lausanne, CH, 2002.

[2] N. Goerke and S. Braun. Building semantic annotated maps by mobile robots. In *Proceedings of the Conference Towards Autonomous Robotic Systems*, Londonderry, UK, 2009.

[3] N. Koenig. Gazebo. http://playerstage.sourceforge.net/index.php, August 2010.

[4] B. Kuipers and P. Beeson. Bootstrap learning for place recognition. In *Proceedings of the 8th National Conference on Artificial intelligence*, pages 174–180, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[5] O. Mozos. *Semantic Place Labeling with Mobile Robots*. PhD thesis, Department of Computer Science, University of Freiburg, July 2008.

[6] Nippon-Signal. FX6 3D Laserscanner. http://www.signal.co.jp/vbc/mems/app_e/item01_2.html, August 2010.

[7] A. Nüchter and J. Hertzberg. Towards semantic maps for mobile robots. *Robotics and Autonomous Systems*, 56(11):915–926, 2008.

[8] S. Oore, G. E. Hinton, and G. Dudek. A mobile robot that learns its place. *Neural Computation*, 9(3):683–699, 1997.

[9] A. Rottmann, M. O. Mozos, C. Stachniss, and W. Burgard. Semantic place classification of indoor environments with mobile robots using boosting. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1306–1311, 2005.

[10] A. Ruckelshausen, P. Biber, M. Dorna, H. Gremmes, R. Klose, A. Linz, F. Rahe, R. Resch, M. Thiel, D. Trautz, and U. Weiss. Bonirob - an autonomous field robot platform for individual plant phenotyping. In *Proceedings of the Joint International Agricultural Conference*, page 464 pp., 2009.

[11] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *Proceedings of the 9th International Conference on Computer Vision*, page 273 pp., Washington, DC, USA, 2003. IEEE Computer Society.

[12] U. Weiss and P. Biber. Plant detection and mapping for agricultural robots using a 3d-lidar sensor. In *Proceedings of the 4th European Conference on Mobile Robots*, pages 205–210, Mlini/Dubrovnik, Croatia, September 2009.