Building Semantic Annotated Maps by Mobile Robots

Nils Goerke and Sven Braun

***Abstract*— The work presented here explains a framework to build semantic annotated maps from laser range measurements of a mobile robot. A hand-crafted and a learning classifier is explained. Two alternative methods to aggregate the resulting class membership vectors into a grid map have been developed and are presented. Both alternative methods will be motivated and described in detail and discussed critically. Results from simulations and real robot experiments will demonstrate the capability and the limitations of this approach to build semantic maps.**

I. INTRODUCTION

A. Motivation

Today mobile, autonomous robots have found their way from being exotic, specialised research platforms to home applications. It is expected, that in the near future the installations of mobile robots in service and domestic applications will increase substantially. The World Robotics Report 2008 states: *"Projections for the period 2008-2011: about 12.1 million units of service robots for personal use to be sold"* from [13].

As a second lucky circumstance the development of laser based range scanners has progressed so quickly that high quality laser measurement is available at a moderate price and a lot of robotic platforms are today equipped with laser range scanners. Thus a wide range of novel applications come into the focus of nowadays robotics research. One challenging goal for the forthcoming robotic research and development is to make robots more autonomous to assign tasks that they can complete on their own.

The job we have in mind for an autonomous robot is to examine an unknown area (e.g. office environment) build a grid map and enhance this grid map with semantic information. One necessary assumption hereby is, that a laser scan can contain enough specific information about the surroundings to yield a robust classification. Recently published work from other research groups indicate that this is possible [15], [16].

Building metric, grid based mappings with the use of a laser scanner while the robot is exploring has been widely reported in the literature and can be regarded as state-of-the art. The family of Simultaneous-Localisation-and-Mapping (SLAM) algorithms are well established in the robotic community since the publication of John J. Leonard and Hugh F. Durrant-Whyte in 1991 [12]. Algorithm repositories and forums for interchanging of data and experience are available and give access to several SLAM implementations [18]. The (typically) grid based metric maps obtained by SLAM algorithms are an extremely usefull basis for further robot control tasks (e.g. navigation, planning, . . .).

Still the representation as a grid map is not always satisfying human communication habits, especially when the operator is not a "robotic specialist", but when the robot is used in a domestic application. Humans typically prefer a linguistic statement instead of a mathematical precise information when interacting with robots: *"The robot is in the seminar room"* is more convenient than the more precise information that the pose of the robot is $(12.40, -3.85, \pi/2)$.

For testing the approach we have chosen to work with the teaching and research mobile robot platform *RoomRider* in a typical indoor, office environment and to use a rather limited set of 5 classes for annotation to represent typical situations of such office environments: doorway, corridor, freespace, room, unknown. The results from the classification are aggregated with two alternative approaches to annotate the grid map, and thus to build the semantic annotated map. The results from simulations and real world experiments in different environments show the capabilities and limitations of the presented approach. Part of this work has recently been published as thesis in computer science [3].

B. Related Work

Classifying laser scans to annotate maps with semantic information has been investigated before e.g. by works of Rottmann [17] and Mozos [16]. Both have used a camera in addition to the laser range sensor to divide rooms in more specific classes, like seminar room, office room, lab and kitchen. They trained strong classifiers with the AdaBoost learning algorithm with extracted features of $360°$ laser measurements. After that, they made a 1-out-of-n decision for the resulting class. To reduce the error rate of the classifiers they use HMM with a transition matrix on the trajectory of the robot.

Different research groups have published methods for detection of one or several of the semantic classes used in this work. In [4], Buschka and Saffiotti describe a virtual sensor for room detection, which can retrieve already visited rooms by features saved in a topological map. In [11], Koening and Simmons developed a doorway detector, that searches for gaps in corridor walls. Althaus and Christensen [1] describe a method for line extraction in sonar date. With those lines they are capable to detect corridors and doorways.

Part of the RoomRider teaching and research robot platform was supported by the Computer Science Students of the University of Bonn under grant SBK-08-01-04-08

Sven Braun has completed his Diploma (MSc) at the Computer Science Department of the University of Bonn, Bonn, Germany brauns@cs.uni-bonn.de

Nils Goerke is with the Department of Computer Science, University of Bonn, Bonn, Germany goerke@ais.uni-bonn.de

TAROS 09, pp.149-156

Towards Autonomous Robotic Systems
Eds.: T.Kyriacou, U.Nehmzow, C.Melhuish, M.Witkowski

II. STRUCTURE OF THE APPROACH

The approach presented here is structured into four major functional sections, see Fig. 1. Each of these sections is subdivided into several subtasks that are necessary to complete the envisaged goal.

1) Acquire data:
   The laser range sensor measures 540 distance values $\mathbf{D}(t)$ in every time step (t) while the robot is moving through the environment.
2) Metric Map:
   Using several successive distance vectors $\mathbf{D}(t), \mathbf{D}(t+1), \ldots$ and the corresponding movements of the robot, the SLAM algorithm GMapping [8] produces a grid map.
3) Classification:
   From the distance vector $\mathbf{D}(t)$, a set of features $\mathbf{F}(t) = \mathbf{F}(\mathbf{D}(t))$ is extracted as basis for the classification. The feature vector is fed into the classifier, yielding the class membership vector $\mathbf{C}(t)$. Each component $c_i(t)$ of the class membership vector $\mathbf{C}(t)$ is the graded belief of belonging to the respective class.
4) Annotated Map:
   The semantic annotated map is constructed by enhancing (annotating) the grid map with the semantic information derived from the class membership vector. Two alternative methods have been developed and tested.

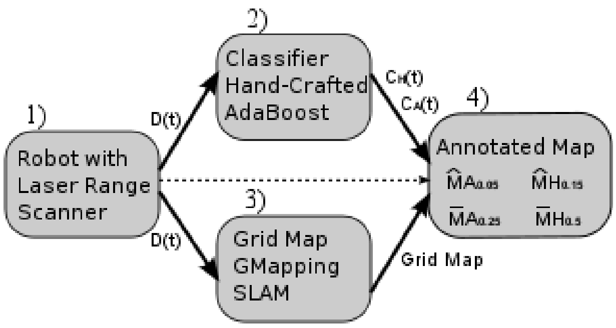

Fig. 1. The approach is structured into 4 major functional blocks: $\mathbf{D}(t)$ distance measurement, $\mathbf{C}_H(t), \mathbf{C}_A(t)$ class membership vectors, annotated maps $\widehat{\mathbf{M}}H_Z, \widetilde{\mathbf{M}}A_Z$, created with the different methods $\hat{1}$ and $\tilde{2}$, and with different spatial resolutions Z.

The complete structure is explicitly prepared to be extended by topological map builder (not depicted in Fig. 1). This is dedicated to build a topological map from the semantic annotated map. Topological maps are said to be even more user friendly and can be the basis for more sophisticated planning algorithms.

III. INFRASTRUCTURE: ROBOT, LASER SCANNER, ENVIRONMENT, GRID MAP

A. RoomRider

We conducted the experiments with the research and teaching robot platform `RoomRider` Fig. 2, developed and constructed by the Department of Computer Science VI, Autonomous Intelligent Systems at the University of Bonn. `RoomRider` is a mobile robot platform based on the consumer product iRobot Roomba® 530 vacuum cleaning robot [10]. The vacuum cleaning robot has been extended by a notebook on top, controlling the robot via serial interface and with a SICK S300® Professional Safety Laser Scanner [19].

Fig. 2. `RoomRider`: the teaching and research platform from the Autonomous Intelligent Systems research group of the University of Bonn, with a Roomba vacuum cleaning robot as basis, a SICK S300 laser range sensor and the controlling notebook on top, photo by courtesy of M.Schreiber.

The SICK S300 Laser Scanner is scanning the area of 270° ($-135°$ to $+135°$) in front of the robot in a height of 21cm above ground with 540 values. The working range for measuring the distances is from 3cm up to 30m. The laser range sensor measures 540 distance values $\mathbf{D}(t)$ in every time step t and transmitts this data via the serial interface to the controlling notebook. The `RoomRider` platform can be controlled with the notebook via the robot middleware `Player` [7] using a slightly modified interface of the roomba 500 driver and the interface for the SICK S300 laser scanner. In addition the simulation environment `Player/Stage` [7] can directly be used to conduct `RoomRider` simulations. In the experiments with the real `RoomRider` we have supervised, and corrected, the movement of the robot (for security reasons) by a human operator. Within the simulations we used a combination of wall-following behaviour and Braitenberg type 3b [2] obstacle avoidance to move the robot. Since the scanning distance of the laser scanner is large enough, any reasonable philosophy for steering the robot (including random walk) is valid, and can be applied as long as the majority of the environment is encountered frequently enough.

B. Environment

We have chosen two real world, office environments called `World9` Fig. 5 and `World10` Fig. 3 as testing ground for the `RoomRider` experiments. For both environments we have created a grid map (see next section) so we could conduct simulations with the `Player/Stage` simulation tool. The size of these environments is `World9`: $46.50m \times 14.25m$ and `World10`: $12.67m \times 12.67m$

Although these environments have a limited spectrum of different situations, the experiments have been conducted during normal office hours. `World10` is smaller in size with just a two rooms, a corridor and a doorway and does not have all situations available: the chairs and tables have been hidden from the laser scanner and thus no chair-legs or table-legs are in sight; the door leaf was kept completely open during all measurements. `World 9` is larger in size, containing several rooms, offices, and laboratories. It has a long corridor with doorways and door leafs, and the rooms contain chairs, tables and all the "normal" equipment that is usual for a typical office floor. Both environments have been mapped and have been used with the robot simulator `Player/Stage`. The classifiers have been developed, and trained with real world data from `World9`.

In addition, further environments (e.g. `World SDR_B` from the *Radish* repository, world SDR site B, [9]) have been used in simulations to test the results in an environment that is different from the environment the classifiers have been developed.

C. Grid Map

The laser range sensor measures 540 distance values $\mathbf{D}(t)$ in every time step t while the robot is moving through the environment; almost every reasonable steering philosophy can be applied. Using several successive distance vectors $\mathbf{D}(t), \mathbf{D}(t+1), \ldots$ and the corresponding movements of the robot, the Simultaneous-Localisation-And-Mapping (SLAM) algorithm GMapping [8] from the robot middleware CARMEN [14] is used to generate a grid map. The map is an occupancy grid map, with a cell size of 5cm$\times$5cm, resulting in 931×286 cells for `World9` Fig. 5, and 254×254 cells for `World10` Fig. 3. All maps had to be reworked manually with a drawing program to close "gaps" that the GMapping produces in areas that have not completely been scanned by the laser beams. A continuous contour is necessary to conduct experiments with the simulation software.

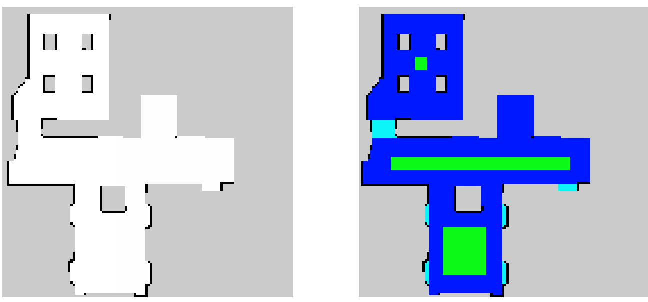

Fig. 3. `World10`: left: result from GMapping; right: grid map annotated with human generated "ground truth" data.

IV. Classification

A. Five Classes

To investigate the capabilities of the approach we have chosen to use a rather limited set of 5 classes that are typical for office environments: doorway, corridor, freespace, room, unknown. These 5 categories are typical for office environments, they are distinguishable from each other and they are meaningful for humans and thus suitable for communication with humans. In fact, we have used 4 + 1 categories; the first 4 categories are to be detected by the classification and the fifth categories which is called *unknown* is getting active in case no of the four categories is detected with sufficient belief. We are convinced that introducing the class *unknown* into the classification is reasonable. Sometimes it is better to be honest, and to state that no believable classification could be found instead of taking the least worse classification result. The assumption that the laser scan contains enough specific information about the office environment is strengthened by recent publications [16]. Including a vision/camera based detection of the different surroundings might be helpful for a more specific classification. Nevertheless, all further processing steps described here apply for a vision based system as well.

1) **D:** Doorway
   A doorway is characterised by a doorframe with a typical width. This width can differ substantially between 70cm to 140cm (site dependent).
2) **C:** Corridor
   Two parallel walls with a minimal length, that have a typical distance (site dependent).
3) **F:** Freespace
   An area of a reasonable size in front of the robot that is not blocked by any obstacle.
4) **R:** Room, office or lab
   Typical for a room is the large number of chair-legs and table-legs. Other room typical characteristics (e.g. rectangular shape, size, ...) have not been envisaged in this work. A further sub classification into different room types has not been regarded to be reasonable by just using a laser range sensor.
5) **U:** Unknown
   The class unknown is applied if the belief for the four other classes is too low.

In contrast to other work, e.g. Mozos [15], [16] where only one category is detected in a pure 1-out-of-n decision process, we allow the classification to result in a real valued vector $\mathbf{C}(t)$ representing the graded belief values of the respective classes. Each component $c_i(t)$ of the class membership vector $\mathbf{C}(t)$ is the graded belief of belonging to the respective class. For each of the four primary classes one classifier is calculating the belief of belonging to this very class as a real value between 0.0 (not at all) and 1.0 (definitely). Thus, it is possible that a situation is belonging to more than one of the four primary classes at the same time: e.g. a doorway with a long door frame can have the characteristics of a short corridor, a wide corridor on the other hand can be correctly classified to be a freespace to some reasonable extent. When the classification shall result in a 1-out-of-n decision, we have to implement a mechanism to perform this: e.g. a winner takes all decision is one possible solution for this. The process of annotating described in this contribution implements two different methods to handle this.

Each classifier has the task to implement a mapping from the 540 dimensional distance vector $\mathbf{D}(t)$ onto a scalar belief value for the very class. All five belief values build the 5-dimensional class membership vector $\mathbf{C}(t) = (c_D(t), c_C(t), c_F(t), c_R(t), c_U(t))$. First tests showed that this mapping can be rather complicated, and we decided to extract sets of features from the original laser scan and use these features as input for the classification. The features have been selected and designed by hand and are in part motivated by recently published work [16]. As an alternative approach not pursued here one could use an automatic dimensionality reduction method (e.g. Principal Component Analysis PCA, Isomap, Vector Quantization, . . . , see [6])

Fig. 4. Three typical doorways with different shape. Clearly visible are some of the situations that can cause difficulties: the door leafs and a second doorway directly behind the first one (most right picture).

B. Hand-Crafted Classifiers

For comparison reasons we have implemented one hand crafted classifier for each of the four classes (**D**oorway, **C**orridor, **F**reespace, **R**oom). Each of this classifiers is extracting some special characteristics (features) from the original scanned distances $\mathbf{D}(t)$ and maps this to a scalar value. The values are normalized to 0.0 to 1.0 and indicate the belief of belonging to the respective class. The class membership vector for the hand-crafted classifiers is denoted $\mathbf{C}_H(t)$.

The Doorway detection assumes that the robot is exactly in the doorframe, with the orientation pointing through the door, that the distance between the objects (shall be the doorframe) detected on the two sides of the robot is between 80cm and 110cm and that there is a free area in front of the robot which is believed if more than 80% of the laser beams in front detect no obstacle in the range up to 100cm.

The Corridor detection assumes that a corridor consists of two straight walls that are parallel and are 170cm to 260cm apart. Two lines, left and right of the robot are fitted through the laser measurements, the deviations from the optimal angle and the allowed distance between the fitted lines is used to calculate the resulting belief value.

The Freespace detection is referring to the $90°$ area in front of the robot. The area up to 40cm in front of the robot must be completely obstacle free, and the the area up to 120cm must be to 90% free. So, if all laser measurements from the $90°$ segment in front of the robot indicate no obstacle closer than 40cm and 90% of them indicate no obstacle up to 120cm then the belief value is set to 1.0.

The Room detection works with the assumption that typical rooms in office and lab environments contain a lot of furniture that can be characterised by detecting and counting the legs. Legs reveal themselves typically by a sudden change (edge) in ad joint distance values. If more than 15 of these edges are detected within the distance vector than the belief for a room is set to 1.0. Unfortunately this policy leads to a lot of false positives, e.g. when a corridor has a lot of corners and door leafs. At the moment we are experimenting with an improved room detection.

All of these hand-crafted classifiers give a 1.0 as belief value, if all described requirements are met perfectly, and a 0.0 if the requirements are violated to a certain extent. The real valued responses are calculated by the grade of match between the laser scan and the requirements. These classifiers have been on line tested and evaluated with the real robot system in the environment `World9` with a human operator steering the robot. The training patterns for the learning classifier origin from `World9`.

C. Learned Classifiers

As a secondary implementation for the classification we have chosen to use a set of learning classifiers based on the AdaBoost method with single layer `9-1`-perceptrons as weak classifiers. We used a set of features $\mathbf{F}_A(t)$ derived from the original laser scan $\mathbf{F}_A(t) = \mathbf{F}_A(\mathbf{D}(t))$ as input for the classifiers. Therefore we have implemented 12 features $\mathbf{F}'_A$ that we expected to contain sufficient information about the environment. We reduced theses 12 primary features to 9 relevant features $\mathbf{F}_A$ by calculating the covariance matrix between all features over a total set of $p = 1 \dots P = 7632$ training patterns $\mathbf{D}(p)$ that have been measured in `World9`. To eliminate features that are too alike we took only those features that have a covariance value below 0.02 and less than 5 values below 0.03.

Below is a short description of the 12 primary features; the features (2,3,6,7,9,10,11,12) that have been selected as being relevant are marked with $*$

- 1 Obstacle free area, calculated by the sum of the distance values $d_i(t)$ over all $i = 1 \dots 540$ laser beams.
- 2^* Mean value of the measured distances.
- 3^* Standard deviation of the measured distances.
- 4 Mean difference of two successive distances.
- 5 Standard deviation of the difference of two successive distances.
- 6^* Minimum of measured distance, shortest distance in scan.

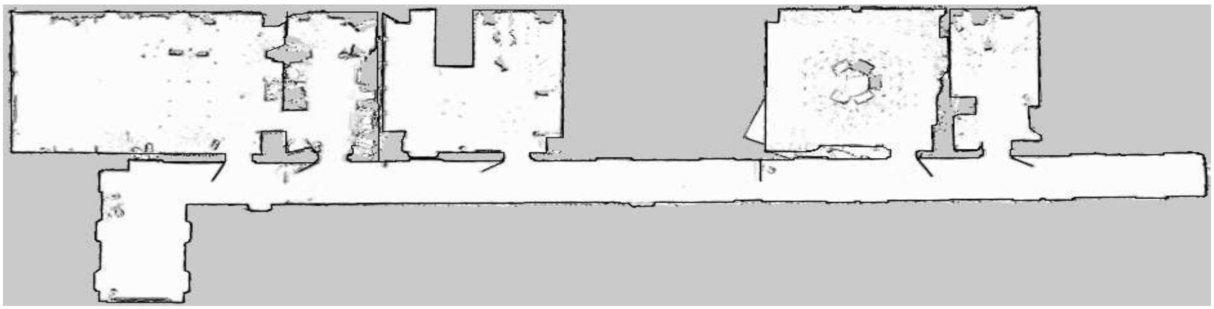

Fig. 5. `World9`, map created with GMapping, containing doorways, a corridor, freespace regions and several rooms, the door leafs are clearly visible, and have not been erased from the map, the cell size is $0.05m$. The map has been reworked manually with a drawing program to get a closed contour for the walls which is necessary for the simulations.

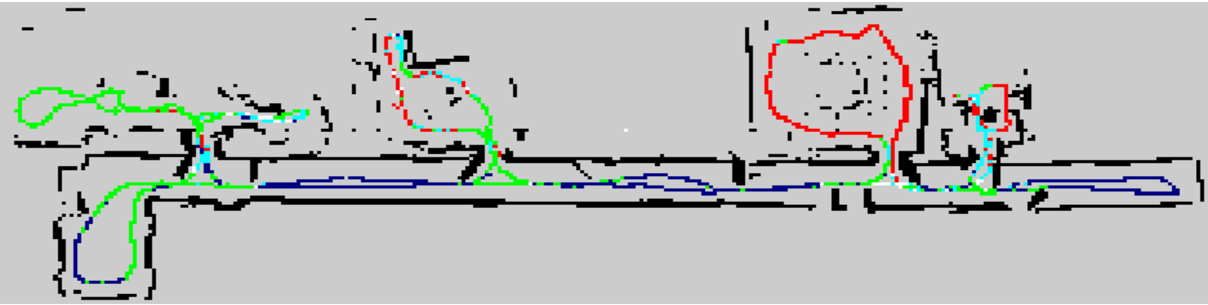

Fig. 6. `World9`, a journey of the real robot with resulting annotations along the path; AdaBoost classification, method 2 for annotation, the cell size is $0.15m$. The different classifications results in the different areas show the feasibility of the approach.

7* Maximum of measured distances, longest distance in scan.
8 Maximum of slope between two successive distances.
9* Number of edges in the whole scan, difference of successive distances is lager than a threshold.
10* Number of relative edges, ratio between two successive distances is larger than a threshold.
11* Distance in the real world between the two closest objects.
12* Angle (with respect to the laser scanner) between these two closest objects.

The AdaBoost algorithm is taking a learning weak classifier as basis and enhances (boosts) the classification quality by taking a next weak classifier with a special focus on those training patterns that have been classified incorrectly before. The final strong classifier is then obtained by combining the sequence of constructed weak classifiers as a weighted sum. Each weak classifier is a `9-1`-perceptron with 9 input neurons and one single output as class belief value. Typically after 4 to 6 boosting steps the results saturated and therefore we have limited the boosting steps to 10. All together $4 * 10$ `9-1`-perceptrons with a total of 400 synaptic weights and 40 AdaBoost weighting factors have been trained for the AdaBoost based classification. The training patterns for the learning classifier origin from `World9`. The class membership vector for the AdaBoost classifiers is denoted $\mathbf{C}_A(t)$.

First experiments in using only one threshold per feature, or a single `9-4`-perceptron for all four classes projecting directly onto the 4 class belief values and the use of multi-layer perceptrons (`9-X-1`, and `9-X-4`, and `9-X1-X2-4`) have started and are currently investigated.

V. Annotation

The classifiers calculate the class membership vector $\mathbf{C}(t)$ in every timestep from the measured distances $\mathbf{D}(t)$ that are said to be typical for that very robot position $(x(t), y(t))$. Annotation means to include the class information for with respect to the robot position into the grid map, which is thereby enhanced from a pure occupancy grid map to a semantic annotated map.

A. Challenges for the Annotation

The spatial resolution of the robot postion $x(t), y(t)$ is in most cases finer than the spatial resolution of the grid map (size of the cells). A cell of the grid map will typically be encountered several times during a journey of the robot, and thus several classification results are obtained for the same spatial cell of the grid. It would be ideal if the classification results would only depend on the spatial location, and all classification results would be identical. But the reality revealed a different situation; deviations between real and detected robot position, changes within the environment between two measurements, fluctuations of the distance measurements, and even the different locations within one grid cell can lead to different classification results for a grid cell. The annotation task has to deal with the different, and sometimes noticeable contradicting, classifications before storing this information into the grid map.

A second challenge arises from the wish to have only one class value per cell in the resulting semantic map neglecting the vectorial character of the classification. If the map has to be annotated following such a 1-out-of n philosophy, and only one single class per grid cell is allowed or preferred, the annotation process has to pay respect to this as well. Two alternative procedures to build up the annotated map have been developed that deal with both requirements at the same time.

B. Method 1: Decide and Aggregate

The first method is directly selecting the resulting class for a time step t from the class membership vector $\mathbf{C}(t)$ using a winner takes all decision, the position information is still on the resolution of the robot positions. The possibly different classification results are now accumulated for each of the grid cells. At the end of the experiment, when all laser measurements have been processed, the resulting final class is again determined by a winner takes all decision. Since a winner takes all decision is allready performed very early, a lot of information that may be usefull has been omitted. The chosen size of the grid cells may influence the resulting map. Annotated maps that have been created using method 1 will be denoted with $\hat{\mathbf{M}}$.

C. Method 2: Aggregate

The second method is paying respect to the vectorial characteristics of the classification process and thus is aggregating all the available information. The real valued class membership values from the class vectors $\mathbf{C}(t)$ are accumulated for all measurements that fall into the respective grid cell.

Only if necessary (e.g. for visualisation) the 1-out-of-n classification is performed to determine a single resulting class per grid cell. Once again the chosen cell size of the grids may influence the result.

D. Annotation using Further Classes

In addition to place the semantic information obtained from the classifiers into the annotated map, the information of the primary laser range measurements and the robot movement can be integrated into the map as well. The laser measurement will leading to the additional class **O**bstacle, and the spatial positions that the robot has not yet visited can be marked as class **N**ot-visited. These additional classes are integrated into the annotated map with respect to the chosen cell size in exactly the same way as the classification results; see Fig. 5 and Fig. 6 for a direct comparison of two maps created with different cell size.

Annotated maps that have been created using method 1 will be denoted with $\widehat{\mathbf{M}}$, those created using method 2 $\widetilde{\mathbf{M}}$. The classifier philosophy (Hand-crafted or Adaboost) is added as letter H or A respectively. The size of the grid (in meter) is added as index. Thus $\widetilde{\mathbf{M}}A_{0.2}$ is the notation for a semantic annotated map with cell size of $0.2m$ that has been annotated using method 2 with the AdaBoost classifier results. Consequently the annotated maps derived with the hand-crafted classifiers using method 1 for the annotation that are depicted in Fig. 7 are named $\widehat{\mathbf{M}}H_Z$ with Z the respective cells sizes from $Z = 0.05m$ to $Z = 0.5m$.

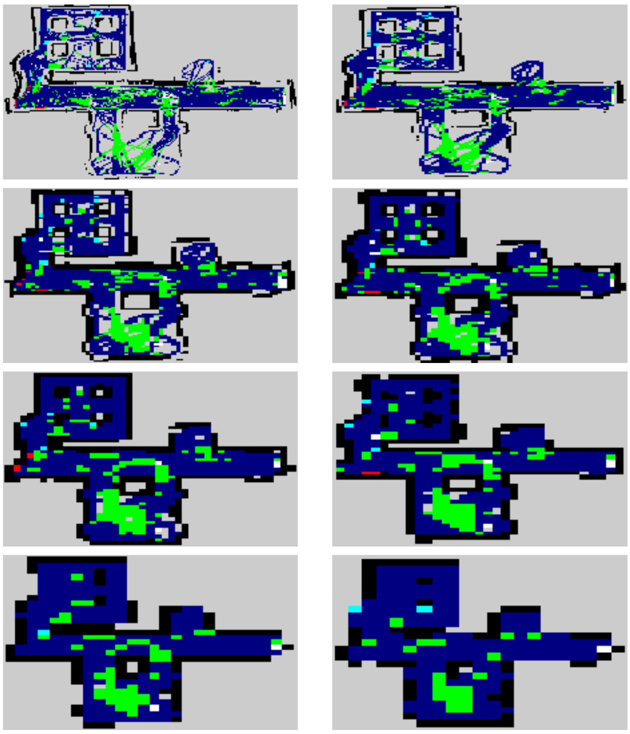

Fig. 7. Semantic map of `World10` created with the hand-crafted classifiers using method 1 for the annotation $\widehat{\mathbf{M}}H_Z$ with Z the respective cells sizes from top left $Z = 0.05m$ to bottom right $Z = 0.5m$, 0.05,0.10, 0.15,0.20, 0.25,0.30, 0.40,0.50.

VI. Experimental Results

To test and validate the presented approach, and to judge the different alternatives for the classifiers and the annotations, several experiments have been conducted using simulations and the real robot.

Some of the results were a little bit disappointing. Especially the large number of false positives in the hand-crafted room classifier that occurred when the robot is definitively located in a corridor. This might have been caused by the large number of edges present in the corridor due to the obstructing door leafs. The other effect that puzzled us, was the rather small detection rate of doorways by both classification schemes. Neither the learned classifier, nor the hand-crafted one showed a satisfactory detection rate for doorways in the real experiments. Up to now, no consistent explanation was found for this effect. We originally had expected, that detecting doorways should be rather easy when looking at the features no. 11 and 12 (since they have been specially chosen for detecting doorways, compare [16]). Unfortunately a reliable detection of doorways is a prerequisite for building valuable topological maps. Detecting corridors and freespace was satisfactory, in Fig. 9 and 10 the freespace in the left part of `World9` has been detected robust. During the journey of the real robot through `World9` Fig. 5 the **R**oom

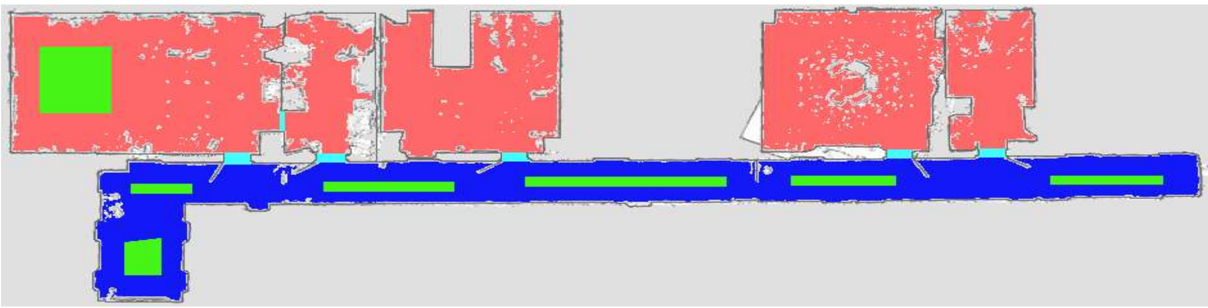

Fig. 8. Hand made Ground Truth data for `World9`, the large room in the right part of the map is a seminar room with a lot of tables and chairs. Only the chair legs and table legs are visible for the laser scanner, thus the room appears rather empty.

characteristics of the large seminar room in the right part of the map has been detected correctly see Fig. 6, as well as the **F**reespace in the left part.

The large room in the right part of the map of `World9` see Fig. 5 is a seminar room with a lot of tables and chairs. Only the chair legs and table legs are visible for the laser scanner, thus the room appears rather empty, although a large hexagonal structure is dominating in the room in reality.

A. Ground Truth

Trying to evaluate the quality of the classification results we found it difficult to obtain ground truth data to compare with. Different persons that we have asked to judge part of the environment `World9` and the complete environment `World10` revealed that even the 1-out-of-n decision for one of the 4+1 classes doorway, corridor, freespace, room + unknown was neither identical, nor stable over time; see Fig. 3 right part of diagram and Fig. 8. Interestingly the classification as *Doorway* was performed almost without any problems and seemed to be easy for the contestants. Perhaps the contestants decision was mostly based on the visual input, which the robot didn't had. Therefore a quantitative comparison with ground truth data has been omitted until stable ground truth data will be available.

B. Hand-crafted vs. Learned

As expected, the difference between the hand-crafted and the AdaBoots trained classifiers are noticeable. One can see some of the effects by comparing the corridor of the resulting maps Fig. 10 and Fig. 6; although the first is a result from the simulation, and Fig. 6 has been produced with the real robot.

C. Cellsize

The influence of the different cell sizes for the resulting semantic maps is large. If the cell size is small, only a small part of the cells would have been visited by the moving robot, and thus gaps would result within the map. Methods that close these gaps will probably resemble the methods we have proposed to aggregate the class membership data. On the other hand, if the cell size is large, too much of the detail information is lost, and the map is probably not specific enough. The cell size we found to be acceptable, (at least for the environments we have tested) was between $0.15m$ and $0.25m$. In Fig. 7 resulting annotated maps $\widehat{\mathbf{M}H_Z}$ with different cell sizes Z are depicted for a direct comparison.

Comparing the two aggregation methods for annotating, and building a semantic map reveals that there are differences between the two methods, but that they are not drastic. We interpreted this as a hint that the developed methods are reliable. Please keep in mind, that for method 2 we have only visualised the results from the winner-takes-all decision. The $\widetilde{\mathbf{M}H_Z}$ maps contain the vectorial information aggregated into the grid cells, available for further processing.

VII. CONCLUSIONS AND FUTURE WORKS

The work presented here is one possible framework for building semantic annotated maps based on laser range measurements from an autonomous robot. Different implementations of the classifiers that detect one of the chosen four classes within an indoor, office environment have been described: a hand-crafted classifier system, and a learning classifier based on boosting perceptrons. Both approaches use features that are extracted from the original laser range measurements to calculate a class membership vector for the robot position. To build an annotated grid map from the different class membership vectors two methods of aggregation have been presented. Results from the experiments conducted in simulations and in real world experiments have been presented.

The presented approaches to build an annotated map from the laser range measurements for a mobile robot showed to be in principle feasible. Although some future work is necessary to make the results more robust and reliable, the presented framework can be applied for building semantic annotated maps by mobile robots. A greater respect should be paid to the choice of the training situations, to further reduce the inaccurate classifications. In total, the idea to prefer the vectorial class membership information over an early 1-out-of-n decision showed to be a valuable approach, although some extra methods had to be implemented. We are convinced, that the presented work is a further step into making robots more end-user friendly.

For direct future work, as follow up developments for the presented approach, we propose to include a camera/vision based classification, a grid map that pays respect to the robot

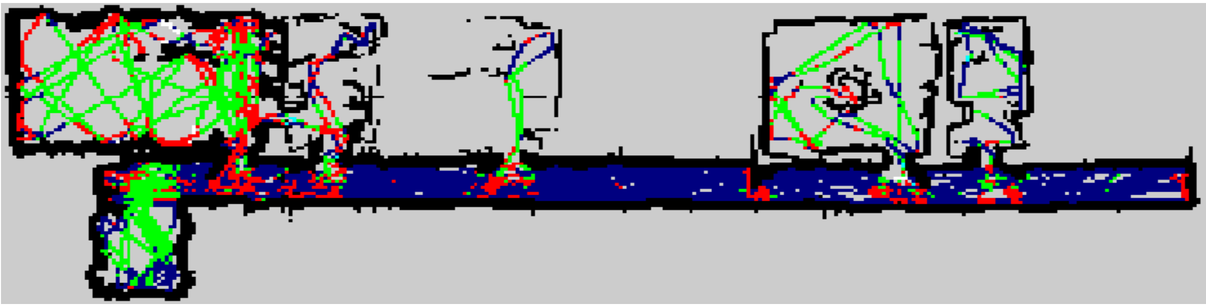

Fig. 9. Annotated map $\widehat{\mathrm{M}}H_{0.15}$ of `World9` obtained from simulations. Depicted is the resulting class from the winner takes all decision.

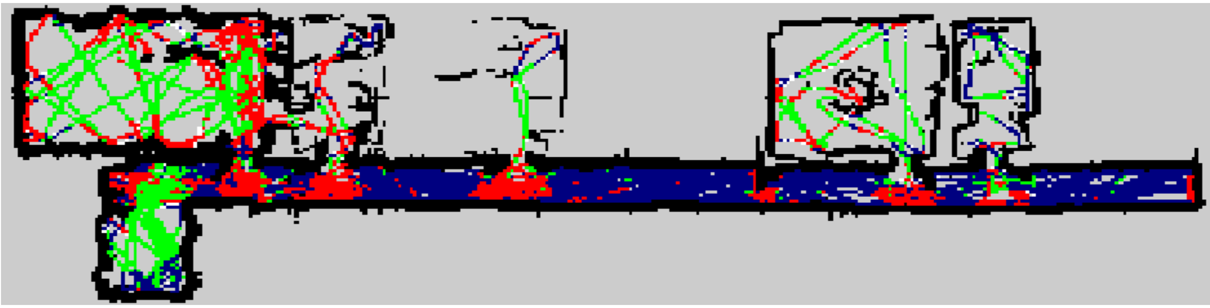

Fig. 10. Annotated map $\widetilde{\mathrm{M}}H_{0.15}$ of `World9` obtained from simulations after a final winner takes all decision for visualisation and comparison. A direct comparison with Fig. 9 reveals that there are differences between the two methods, but that they are not drastic.

orientation and a more reliable doorway classification. When the doorways can be detected more reliable the way to realize a topological mapping is open, since doorways typically connect rooms with each other and rooms with corridors. This structural property of office environments will then be used to build topological maps.

A challenging project for the future would be to make the class definition and the class specification the result of a psychophysical voting by possible human users of autonomous service robots in home environments.

REFERENCES

[1] P. Althaus and H. I. Christensen, *Behaviour Coordination in Structured Environments* , Technical Report Centre of Autonomous Systems, Royal Institute of Technology (KTH) Stockholm, 2003.

[2] V. Braitenberg, *Vehicles: Experiments in Synthetic Psychology*, 1984, The MIT Press, Cambridge.

[3] S. Braun, *Erzeugung von semantisch annotierten Karten von Büroumgebungen*, Diploma Thesis (MSc) (in German language), Department of Computer Science, University of Bonn, Bonn, 2009.

[4] P. Buschka and A. Saffiotti, *A Virtual Sensor for Room Detection* , In Proc. of IROS, IEEE/RSJ Intenational Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 2002

[5] T. H. J. Collett and B. A. MacDonald and B. P. Gerkey, *Player 2.0: Toward a Practical Robot Programming Framework* , In Proc. of ACRA'05, the Australasian Conf. on Robotics and Automation, Sydney, Australia, 2005.

[6] I. K. Fodor, *A Survey of Dimension Reduction Techniques*, Technical Report UCRL-ID-148494, Lawrence Livermore Nat'l Laboratory, Center for Applied Scientific Computing, June 2002.

[7] B. P. Gerkey and R. T. Vaughan and A. Howard, *The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems*, In Proc. of ICAR'03, International Conference on Advanced Robotics, University of Coimbra, Portugal, 2003.

[8] G. Grisetti and C. Stachniss and W. Burgard, *Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters*, IEEE Transactions on Robotics, 2006, 23(1), pp. 34-46.

[9] Andrew Howard and Nicholas Roy, *The Robotics Data Set Repository (Radish)* , Official Website: http://radish.sourceforge.net, last checked: 20 April 2009.

[10] iRobot, *iRobot Corporation: Home Robots* , Official Website: http://www.irobot.com, last checked: 15 May 2009.

[11] S. Koenig and R. G. Simmons, *Xavier: A Robot Navigation Architecture Based on Partially Observable Markov Decision Process Models* , In Book D. Kortenkamp and R. Bonasso and R. Murphy,ed., Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems, 1998, The MIT Press, Cambridge, pp. 91 - 122.

[12] J. J. Leonard and H. F. Durrant-Whyte, *Simultaneous Map Building and Localization for an Autonomous Mobile Robot*, in Proc. of IROS'91, IEEE/RSJ International Workshop on Intelligent Robots and Systems, 1991, vol.3, pp. 1442-1447.

[13] G.Litzenberger, *World Robotics 2008: Service Robots: assistants in private and professional life*, Press release, IFR Statistical Department, 15 Oct 2008, Frankfurt Germany.

[14] M. Montermerlo and N. Roy and S. Thrun, *Perspectives on Standardization in Mobile Robot Programming: The Carnegie Mellon Navigation (CARMEN) Toolkit* , In Proc. of IROS'03, IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, United States of America, 2003.

[15] O. M. Mozos and W. Burgard, *Supervised Learning of Topological Maps using Semantic Information Extracted from Range Data*, In Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 2006.

[16] O. M. Mozos, *Semantic Place Labeling with Mobile Robots*, PhD Thesis, University of Freiburg, Germany, 2008.

[17] A. Rottmann, *Bild- und laserbasierte Klassifikation von Umgebungen mit mobilen Robotern (in German)* , Diploma Thesis (MSc) Computer Science Department of the Albert-Ludwigs-Universitt Freiburg, Freiburg, 2005.

[18] C. Stachniss and U. Frese and G. Grisetti, *OpenSLAM.org is a platform for SLAM researchers which gives them the possibility to publish their algorithms*, http://www.openslam.org, 2008.

[19] SICK, *SICK.com Homepage* , Official Website: http://www.sick.com, last checked: 15 May 2009.