in: Proc. 55th Int. Scientific Colloquium, Ilmenau, Germany, 2010, pp. 294-299, Verlag ISLE 2010

WHAT IS DIFFERENT? - MODELING THE CHANGEABILITY OF THE ENVIRONMENT

Jens Kessler, Christopher Gaudig, Christof Schroeter, Horst-Michael Gross

Ilmenau University of Technology
Neuroinformatics and Cognitive Robotics Lab
jens.kessler@tu-ilmenau.de

ABSTRACT

Within the field of mobile robotics, mapping is a well known problem to be solved to enable a mobile system to drive autonomously. In this work we present an improved approach on dynamic map adaptation with a previously given, static map, built with a laser scanner. We are focusing on modeling changes within this map at different time scales, so old states of the environment are not getting lost instantaneously. We discuss this problem under the consideration of a) keeping the map consistent while including changes and b) avoiding the conflict on fast adaptation vs. forgetting important environment structures. To model these properties we built a stack of maps considering different time levels. In each stack observation data are included with different insertion rate, and statistics of laser beam pairs are built to guarantee consistency of the estimated map. This allows us to construct a view of the change of the environment over time, which classical SLAM approaches cannot do.

***Index Terms*—** Mapping, dynamic map adaptation, statistical map learning

1. INTRODUCTION

In mobile robotics the tasks of finding an appropriate way to drive to a goal, following a person, or even to localize the robot within the environment are still challenging. To enable a mobile robot to act well within the environment, the robot needs an internal representation of the environment, usually called a map. The problems of interaction with the environment arise by using an outdated representation (map) of the environment and releasing actions to an actual environment, e.g. by trying to drive through a blocked floor, a closed door or by trying to drive through moved pieces of furniture. Thats why the robot has to adapt the internal map in a consistent way by including new states of the environment and also retaining the old states of the environment. The type of map used depends on the sensor used to observe the environment. In our case the sensor is a laser scanner with a measurement range of 270°, so all observations are pairs of range measurements coupled with viewing angles. From such scans maps can be constructed. In literature the problem of map building was discussed extensively in the past ten years (see section 2 for details). The main goal of all approaches was the automatic construction of maps within an initially unknown environment. This class of approaches refer to the SLAM problem definition (**S**imultaneous **L**ocalization **a**nd **M**apping) and assumes a static map, which means the state of the environment does not change over time. The state of the environment as well as the successive states (poses) of the robot are unknown and have to be estimated. Our approach differs from that assumption in a way that we do not assume the state of the environment as constant over time and we assume the pose of the robot within the environment is known. An environment state is for example the probability of a cell being an obstacle or in our case the value of a hypothetical distance to the next obstacle from a defined point, which is in our case the center of a map node. We also assume that a reliable state cannot be obtained after one observation, but that only repeated observations can guarantee a stable long term state.

By partially observing the environment we can acquire new information of the current state of the environment, not knowing at that point in time if the current observation is the common state or an outlier. So our main goal is to detect a reliable distance to the next obstacle from a defined spatial point. To create a map of the environment, we construct a set of nodes, where the centers of these nodes are the reference points of possible obstacle configurations. These nodes contain information coded in radial coordinates. So each node can present a state, storing for every angle the most likely distance to the next obstacle. This distance per angle does not only depend on observations collected at that angle, but is also correlated to distance measurements of neighboring angles to provide additional information on the most likely state. This is also helpful since the state of the environment can only be partially observed, because laser scanners do not always cover the full scanning circle, and the full environment state can only be assembled by many observations of different parts of that environment. So the goal of our approach is to collect observations and to model for each node of the map *how observations may belong together* to get an assumption on the *usual* state of the environment. Also the time (or the number of observations) until a "fact" of the environment is included into the internal world representation is an important property of world modeling. We wanted to construct an actual model of the environment, with the last taken observation included to use in obstacle avoidance or person tracking, and we wanted a long term stable representation to slowly adapt on world changes and include only the stable (immobile) parts of the environment. The mechanism of the long term memory, its stability and slow adaptation is a key point on robust navigation in terms of path planning and localization and is a key feature of our proposed algorithm. To model these different adaptation requirements we create different "time levels" per node, each representing a different level of adaptation speed.

This paper is structured as follows: in the next section we

This work has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 216487 (CompanionAble Project).

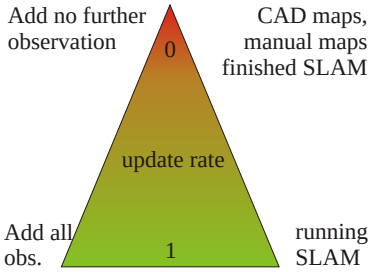

Fig. 1. *A scheme of update rates. While most algorithms assume statical maps they either include no new information into that map (when they assume the map is complete) or all information (when map building is in progress). Only a few algorithms, including that one presented her, use a different update rate to include and forget information regarding the map.*

present the state of the art within the field of map building and updating. In the following section we discuss how data averaging can be applied to different time levels of memory, like short term memory, mid term memory and long term memory and how the map is structured in order to allow updates. In the fourth section the description of the problem of consistent state estimation is given, followed by the key ideas on dynamic map adaptation. In the fifth section experiments and their results are shown. The paper is closed by a short reflection on the presented methods and what further work has to be done on this field of research.

2. RELATED WORK

At the beginning of navigation research in robotics, maps where constructed entirely by humans, since other basic problems where in the focus of operation [1]. As these problems of localization and robot motion where solved to a certain degree, also the problem of mapping came into focus of research. Soon it was understood that the learning of a map of an unknown environment was closely connected to the estimation of the pose of the robot within that environment [2]. The map estimation and the pose estimation are in fact closely coupled estimation problems. With this knowledge a class of SLAM algorithms evolved over the following years [3],[4],[5],[6],[7]. The drawback of all these algorithms is the assumption of a static environment, since most of these algorithms try to find a match of laser-scans [6],[7] or landmarks [3],[4],[5], which assume a constant position of these features within the map over time. If that assumption fails, it is not possible to assume a correct position of the robot and so the whole SLAM solution will fail. All these approaches of classical SLAM have in common that they represent just one snapshot of the environment, representing only the last visited state of the environment and assuming this state is valid all time. So one has the choice to include every new observation into the map, when the SLAM algorithm is running and the needed area to be mapped is not covered, or to include no observation into the map, when the needed area is covered (see Fig. 1). Nevertheless in the past two years various approaches have evolved to overcome these constraints and could update maps for an entire system life cycle. These techniques still require a correct localization hypothesis when returning to a previously visited place [8] or at least some remaining valid landmarks to retain a correct position [9]. When some correspondences to previously seen landmarks are present, newly seen landmarks can be added to the scene, or not seen landmarks can be removed from the scene description. It is the first time this new class of scene description deals with a memory model to filter information before that information becomes part of the scene or is forgotten. It is achieved by simply counting the appearance of visual features, residing in the short term memory, and adding features to the scene when a certain counting threshold has passed. When a feature should be seen at a certain position but is missing for a certain number of observations, this visual feature is removed from the scene description. As [10] mentioned, this is biologically plausible, since Atkinson found three kinds of time dependent visual memory systems within human brains: a short term memory for sensory details with an average time span to forgetting of under 1 second, a reduced mid-term memory with an average time span of 15-30 seconds, and a long term memory, where only repeated information from mid-term memory is stored, which guaranties a very stable memory representation over time. This knowledge is often used implicitly when dealing with maps. For example [11] tried to classify sonar scans as ”moving objects” or ”static background” before mapping, than only the static measurements where used for mapping. On the other hand [12] used the same classification to separate humans from the background to get useful person hypotheses.

Like in the work of [9], which deals with adapting a map containing visual features by using time dependent memory, the approach of [13] presents another idea on map adaptation, based on laser scanner observations, to reflect the changeability of the world with a time dependent memory model. Here not a simple counting threshold is used to select laser scans as the valid observation, but a statistical model, derived from a memory representation (for details, see section 3). A map is built with nodes labeled with laser scans at the position of that node. These scans are independent from each other and represent each the most likely scan observed at a particular angle. With a scheme on including new scans to that particular angle and forgetting scans they where able to model different types of time memory. This work is the foundation of our approach. A main drawback of this method is the dependency of the environment state only on the observation count of distances per angle. This can lead to inconsistencies since each angle is handled independently and the result of the angles state estimation depends only on how often it was observed and not on states of neighboring angles, which give additional information on the decision of the most likely state. In [14] the author suggests that statistical inference of observed cells describe the state of neighboring unobserved cells, because the system has learned before that these cells behave equally when changing because they belong to the same object. So a door state could be estimated by just observing a part of that door and by statistical knowledge previously collected. This is a property [13] have not taken into account. Both ideas presented in [13] and [14] form the basis of the work presented here and we fuse both ideas on coding maps as most likely previously measured ranges per angle bin, and deciding on a state of one entity (here a range per angle bin, in [14] the state of an occupancy cell) by using neighboring state estimates.

3. DIFFERENT TIME LEVELS OF MEMORY

The first question is: how to implement different time dependent memory characteristics into a model of state estimation? A common way is to take a sliding window averaging

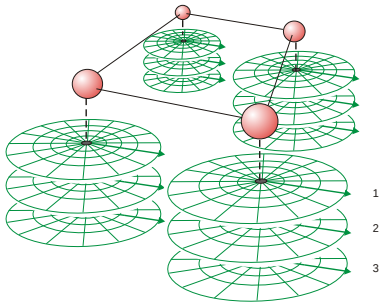

Fig. 2. *Different time levels of the map are represented by nodes with different λ_i parameters. Each node level consists of a set of angle bins, collecting a set of distance measurements in their corresponding angle interval. Additionally each angle bin is split into range intervals to collect correlation information from ranges measured inside the current and the neighboring angle bin.*

of the form $d_{t+1} = (1-u) \cdot d_t + u \cdot d_{curr}$, where d is in our case a range measurement at one angle. One sample d_{curr} would then have a weight (or influence) on the result of $w(t) = u \cdot e^{ln(1-u)\cdot t}$. After a time of $t_{1/2} = \frac{ln2}{\lambda}$ (with $\lambda = -ln(1-u)$ and $u = e^{\lambda} - 1$) this sample would have lost half of its initial averaging weight. The value of λ regulates the level of forgetting old measurements and respecting new measurements. In [13] it was shown, that the decay rate of $w(t)$ could be experimentally proven in biological organisms also. However, this averaging procedure has the problem that it can produce results of d never measured by the laser system and tends to fade from one dominant measurement to the next. One requirement of the estimation of the most likely d is that only previously measured values should be chosen. So the averaging process has to be interpreted as a selection process from a *set of measurements*. In this fashion the parameter λ regulates the interval of inserting new data and releasing old data from that set and also the amount of data to insert and forget (see [13]). By randomly choosing the data to insert from the incoming measurements and randomly forgetting old measurements the memory model is established with different time characteristics. So the designer can decide with the parameter λ which forgetting rate is appropriate to construct short-term, mid term and long term memories.

The map consists of a set of nodes, where each node has different instances (parameters λ) to construct a different memory level for the same spatial space(see Fig.2). Each time level of a node consists of a set of angle bins, covering the full 360° range. In these bins observations to the next measured obstacle are collected in a unsorted set. Additionally each angle bin is also split into range bins. These bins are needed to find the most likely distance in that angle bin with taking into account all measurement from the neighboring angle bin. Each distance measurement belongs to a range bin. The occurrence of range-bin pairs is collected to guarantee a consistent overall estimation, which is described in detail in section 4. In each time level the parameter λ is chosen for a different time characteristic, like short term, mid term and long term memory. This parameter influences the decay rate of the sliding window (or the length of that window) or, interpreted as an insertion rate, the number and interval of new observations to be inserted and old observations to be removed from the current set of observations. To remove a range measurement from the current set, we also have to update the statistic on the correlation of the current ranges to the neighboring ranges. Each range measurement is connected to a so called co-occurrence pair, which is described in more detail in the next section. These pairs influence the co-occurrence count in a matrix. Here it is important to mention that upon removal of a measurement from one angle bin, also a randomly selected co-occurrence entry from the same angle bin and the corresponding range bin is also removed. So an overall map with different levels of consistent more or less actual states will occur. The benefit of such a map lies in a measurement of path planning stability, where paths at different time levels may differ from each other, and also in an improved localization capability, since different world configurations can be assumed to be valid.

4. THE PROBLEM OF CONSISTENT WORLD REPRESENTATION

As stated in section 2, we assume that the pose of the robot in map coordinates is always known. So we concentrate only on map adaptation and do not solve a SLAM problem. To formulate the problem definition on consistent world states we have to describe some basic structures first. We create nodes that form the map, labeled with virtual 360° scans with configurable angle resolution (see Fig. 3). Each angle interval is called an angle bin, denoted as S_a. Each node has its first angle bin viewing into direction $\phi_n = 0$. Each angle bin stores a collection of virtual range scans. To insert such a range scan into an angle bin we have to transform an actual robot scan into nodes coordinates, since the robot usually is not at the nodes position.

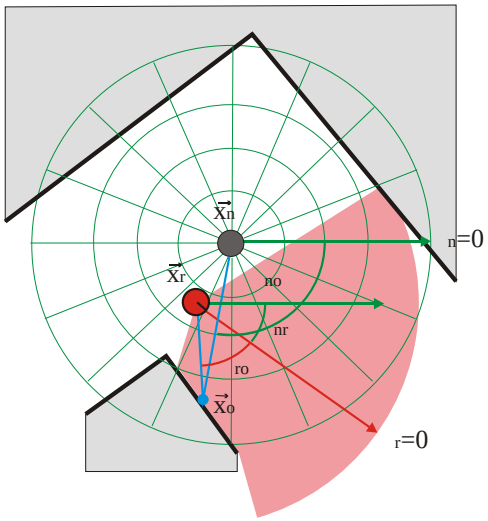

Fig. 3. *A node with 16 angle bins and 4 range bins per angle. The scan taken from the robot has to be transformed to the node's position. Each point of the robot's scan is transformed into global Cartesian coordinates and afterwards to polar coordinates relative to the node's center.*

As shown in figure 3, the robot has its own local coordinate system, where scans are stored relative to the robots viewing direction ϕ_r. The pose $\vec{x}_r$ of the robot is also known as well as the view direction ϕ_{nr} in global coordinates. The transformation is executed in two steps. First the position of the obstacle point $\vec{x}_o$ is calculated in Cartesian coordinates relative to the robot's position $\vec{x}_r$ and the global view direction ϕ_n.

$$\begin{bmatrix} x_{ro} \\ y_{ro} \end{bmatrix} = \begin{bmatrix} d_r \cdot cos(\phi_{nr} + \phi_{ro}) \\ d_r \cdot sin(\phi_{nr} + \phi_{ro}) \end{bmatrix} \quad (1)$$

with d_r the measured distance to the obstacle.

In the second stage we simply add the distance between $\vec{x_r}$ and $\vec{x_n}$ as offset to the coordinates x_{ro}, y_{ro} and transform these coordinates back into polar coordinates.

$$\begin{bmatrix} x_{no} \\ y_{no} \end{bmatrix} = \begin{bmatrix} x_{ro} \\ y_{ro} \end{bmatrix} + (\vec{x_n} - \vec{x_r})$$
$$\begin{bmatrix} d_{no} \\ \phi_{no} \end{bmatrix} = \begin{bmatrix} \sqrt{x_{no}^2 + y_{no}^2} \\ atan(\frac{y_{no}}{x_{no}}) \end{bmatrix} \quad (2)$$

Here d_{no} and ϕ_{no} are the distance and angle of the virtual scan from the node. We can define that such a pair of obstacle angle and obstacle distance is an observed state of the environment. In [13] the angle ϕ_{no} is discretized into bins and all values d_{no} are collected within these bins. By selecting the median of all stored range values a "most likely" scan is selected to represent the state of the environment at that angle. The benefit of this selection is, that only previously measured values are possible. But this method has one major drawback, because all angle bins are handled independently. So the selection of the "most likely" measurement is only based on the measurement count *within that angle bin* and is not related to neighboring bins. This can cause inconsistencies of "half visible" objects in impossible states (half in new position and half in old position), when one part of the object is seen more often in the new state than the other part.

As a solution to this problem we take into account the statistical dependencies of neighboring scans, since states of real objects within the world do influence neighboring scans if they span over several angle bins. To capture this influence we model so called co-occurrences. This means that a measurement in angle bin A is correlated with a measurement in the neighboring angle bin B (see Fig. 4). For simplified modeling we take the Markovian assumption that a scan is only correlated with its immidiate neighbor scan. We built a simple statistic upon this fact by also clustering the laser scans into range bins (per angle) and counting the co-occurrence to the neighboring scan in a co-occurrence matrix. In fig. 4 for example, whenever a scan in angle bin A was measured in range bin 3, the range bin of the next scan B was 2. The same holds true for range bin $\langle A, 4\rangle$ correlated with range bin $\langle B, 4\rangle$ of the next angle bin. The update of the co-occurrence matrix is quite simple: every time a scan (classified into range bin α) has a valid next scan (classified into range bin β), the co-occurrence matrix of angle bin $S_a[\alpha][\beta]$ is increased by one.

Note that these scans and these statistics are collected over time, so many possible configurations of the environment are observed. Also only parts of what may be seen from the position of the node is observed at a time, since laser scanners usually do not cover the full circle at one observation. Since we had to consider only a predefined length of observation time of the sliding window, we restrict the co-occurrence matrix to a maximum entry count. So, when the full count of possible entries is reached and a new observation is added to that angle bin, we randomly clear one entry from the matrix and also remove the corresponding scan in the range bin. Finally, a list of measured ranges per angle bin and a co-occurrence matrix per angle bin results, which represents the statistical correlation between the range bins of the actual scan and the neighboring scan. The interesting question to solve now is "What is the most likely range bin configuration

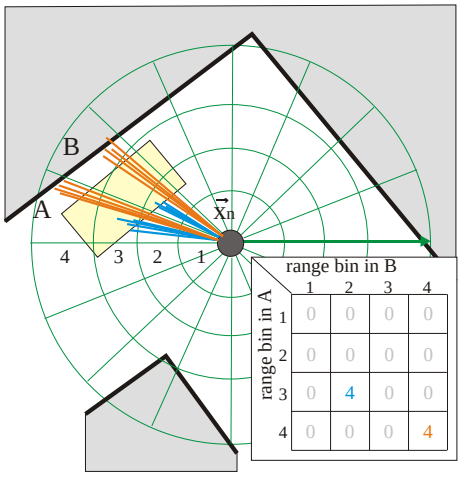

Fig. 4. *Example of one of 16 transition matrices, containing statistics regarding range scans from angle bin A to B. Here two configurations exist: one tells that scans in range bin 3 are always followed by scans in range bin 2 (blue), the other that scans in bin 4 are followed by scans in bin 4 (orange).*

for all angle bins at that node?". This represents the assumed actual state of the environment at that position. To solve this question we have to combine all available statistic data to find the configuration with the highest probability. Let the scan distance of angle bin S_i be d_i and the scan vector of all scans be $\vec{D} = \langle d_1, d_2, ..d_n\rangle$, then we can formulate the estimation problem as follows:

$$p(d_1^*, d_2^*, ..., d_n^*) = max_{\vec{D}}\, p(d_1, d_2, ..., d_n) \quad (3)$$

where $p(d_1, d_2, ..., d_n)$ is the probability distribution over all scan ranges and $\langle d_1^*, d_2^*, ..., d_n^*\rangle$ the optimal combination of scans. By applying the Markov assumption we can rewrite

$$p(d_1, d_2, ..., d_3) = p(d_1) \cdot p(d_2|d_1) \cdot ... \cdot p(d_n|d_{n-1}) \quad (4)$$

From this distribution the most likely combination of the ranges $\langle d_1, d_2, ..., d_3\rangle$ has to be found. This problem can be visualized as a pseudo cyclic path search through a field where each node represents a range bin within an angle bin (see Fig. 5). Pseudo cyclic means that the first row of nodes and the last row of nodes are identical and a valid path has to end at that range index it has started from. The weights of the graph edges are the conditional probabilities $p(d_n|d_{n-1})$ which could be easily acquired from the co-occurrence matrices. The starting costs for the nodes of the first row is the corresponding prior $p(d_1)$. The path with the *maximal product* of all edges is our required result. Obviously paths starting with a prior of zero can be neglected. Additionally, as seen in Fig. 5, we have to consider only these paths where a no-zero weight to the next angle bin exists. Normally the amount of possible paths is not very large, so it is possible to search through all these paths with brute-force, but there is another way to gracefully solve the same problem. The search of an optimal path is also a well known problem in informatics. Dijkstra proposed such a solution in [15]. The difference to our graph is, that a path is found where the *sum of all weights is minimal*. But we can easily reformulate the graph weights to create a graph suitable for Dijkstra graph search. We simply invert the graph weights to new weights $w_{ij} = 1/p(d_j|d_i)$ (see Fig. 5 c)) and also transform the prior in the same way. In this fashion we can effectively calculate the optimal path even in complex scenarios with a high count of possible paths.

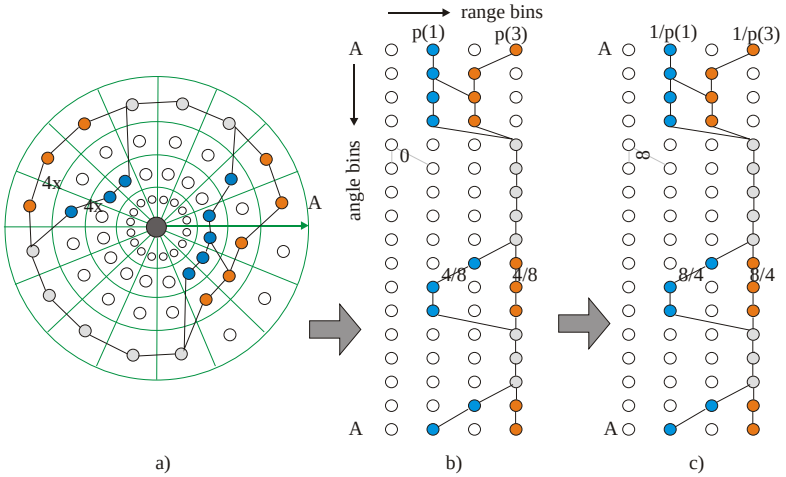

Fig. 5. *The transformation from the bin representation to a graph. Only these values in the transition matrix from one angle bin to the next create edges, which have a co-occurence count greater than zero (a) and b)). The weight of the graph is the number of visits in the range bin divided by the overall count of visits in the angle bin. To reformulate the problem of finding the maximum product we create a graph with reciprocal weights and now can search for the path with the minimum sum. Columns of the graph are the range bins, rows of the graph represent the angle bin. Note that the number of angle bins is one bin extra to copy the first row into the last row and create a pseudo cyclic graph.*

With that path we have an optimal range bin for every angle bin. From each optimal range bin we select the median range measurement, like it is done in [13], to be sure to select a distance measured before. The only exception to consider are angle bins without any measurement (because we have never observed anything within that bin). In this case all ranges are assumed possible and have an entry inside the co-occurence matrix, but the weights to the next angle bin are set to one. So all paths passing that bin are equally possible.

5. EXPERIMENTS AND RESULTS

In this section we show a short proof of concept with some experiments. Within these experiments we evaluate only one node of the map to give a better impression of side effects. Of course all shown properties can be extended to the set of all nodes defining the map. To show the properties we have placed the robot onto a floor of our lab. The space initially observed is a circular area of 3 meters in diameter. The scans are incomplete since the robot is only equipped with a scanner covering $270°$. The scan is transformed to a scan at the node's position with bins of one degree resolution. Each angle bin is equally partitioned into segments of 7,5 cm, so each angle bin consists of 20 range bins. Figure 6 shows a result on convergence from one state of the environment to the second state. The remaining (unlikely) scans are shown as light red fillings of the range bins.

The resulting ranges switch from one state to the other when enough observations are collected and old observations are forgotten, so the probability of range bin $S_{i+1}(r_{new})$ after range measurement $S_i(r_i)$ is higher than the old range bin $S_{i+1}(r_{old})$. In this case (enough) observations of only a part of the obstacle lead to a jump in scan prediction. This can be seen in Fig. 7, where the robot has previously observed the full new obstacle a few times and afterwards only is able to observe only a part of the obstacle (here the robot has turned and could only observe the "upper" part of the obstacle). Note that now previously unseen parts of the environment could be observed and are now included immediately into the most likely scan.

A last experiment is dedicated to position jumps due to a

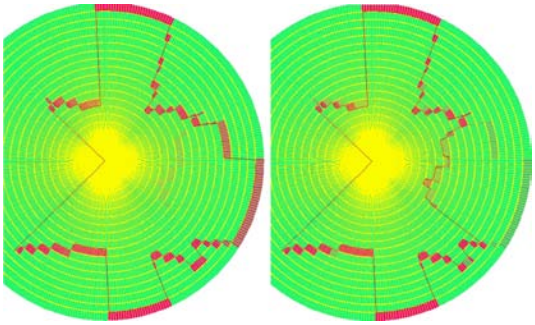

Fig. 6. *Adaptation of one time level of one node. On the left is the state as observed usually before. Then the environment changes and the set of measurements and the co-occurences slowly adapt to the new state until most observations belong to the new state and the estimated state result (red line) changes to the new state.*

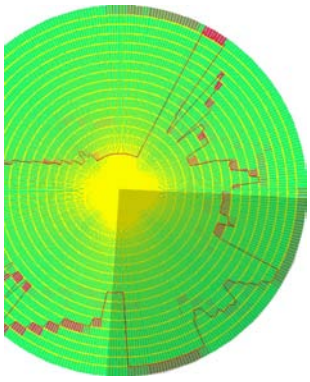

Fig. 7. *The gray part could not be observed by the laser for a long time. But both possible states where observed before and by assuming the state of previous angles (observed regularly) the most likely states of the dark gray angle bins could be estimated depending on the previous states.*

bad localization of the robot. In such a case the transformation of the robot scan will result in an incorrect virtual scan with an error in translation and rotation. This can be seen in Fig. 8. When the "wrong" observation occur more often, the resulting scan begins to focus also on the transformed scan with the most frequent localization error. Nothing happens when de-localization occurs only for a few observations.

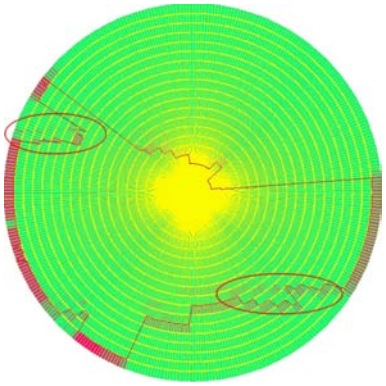

Fig. 8. *The effect of localization noise. Here two hypotheses are measured because of an wrong scan transformation. The dominant hypotheses is selected.*

Last experimental results discuss the processing time. Since only one scan could be included into one node at a time, an update step of the full map consists only of that time. So processing time of the update of one scan is equal to processing time of the whole global map! Listed below are the averaged processing time results split to the number of paths that have to be searched because of different counts of starting points with non-zero prior.

Number of start bins	processing time
1	17 - 31 ms
2	43 - 52 ms
3	72 - 81 ms
4	90 - 98 ms

Table 1. *Table with number of start range-bins in the first angle bin and the needed processing time. As can be seen the processing time is nearly linear.*

Here we have an average of 24 ms on each path run. When all 20 ranges of the start bin contain a valid start point (because they where measured at least once), we will need 500 ms. So on the worst case scenario we can update the map with two scans per second. The results where calculated with a dual core processor running at 2.4 GHz.

6. CONCLUSION

The benefit of our approach is the representation of consistent expected observations at different levels of time-scale, so no mixtures of observations which are very unlikely to occur in the same state are selected. All this information is represented in a consistent fashion to the mobile robot to improve localization and path planning.

We take into account different time levels by assembling a map containing nodes where each node of the map collects sensor information with a different update rate. So we can model maps with a long term memory characteristic, short term memory characteristic and all in between. Our approach also presents how to create consistent environment information when the impression of a certain place was only partially observable and collected at different points in time. Future work has to be done to further increase the performance of the optimization task on finding the optimal scan configuration. Also the side effects on localization errors have to be investigated. As said before path planning and localization can benefit from different time scales. This also has to be evaluated further.

7. REFERENCES

[1] S. Thrun, M. Bennewitz, W. Burgard, A. Cremers, F. Daellaert, D. Fox, D. Haehnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Minerva: a second-gerneration museum tour-guide robot," *IEEE Conference on Robotics and Automation*, vol. 3, pp. 1999–2005, 1999.

[2] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, The MIT Press, 2006, chapter 10.

[3] R. Smith, M. Self, and P. Cheeseman, "A stochastic map for uncertain spatial relationships," *Robotics Research, The Fourth Int. Symposium*, pp. 467–474, 1988, The MIT Press.

[4] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *International Journal of Robotics Research*, vol. 23, pp. 693–716, 2004.

[5] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam: A factored solution to the simultaneous localization and mapping problem.," in *Proceedings of the AAAI National Conf. on Artifical Intelligence*, 2002, pp. 593–598.

[6] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, pp. 34–46, 2006.

[7] Sprickerhof J., A. Nuechter, K. Lingemann, and J. Herzberg, "An explicit loop closing technique for 6d slam," in *In Proceedings of the 4th European Conference on Mobile Robots*, 2009, pp. 229–234.

[8] A. Koenig, J. Kessler, and H-M. Gross, "Improvements for an appearance-based slam approach for large-scale environments," in *In Proceedings of the 4th European Conference on Mobile Robots*, 2009, pp. 235–240.

[9] F. Dayoub, T. Duckett, and Cielniak G., "An adaptive sperical view representation for navigation in changing environments," in *In Proceedings of the 4th European Conference on Mobile Robots*, 2009, pp. 1–6.

[10] R.C. Atkinson and R. Shiffrin, "Human memory: A proposed system and its control processes," *The psychology of learning and motivation*, 1968, chapter 2.

[11] D. Haehnel, R. Triebel, W. Burgard, and S. Thrun, "Map building with mobile robots in dynamic environments," in *In Proceedings of IEEE int. Conf. on Robotics and Automation(ICRA)*, 2003, pp. 1557–1563.

[12] R. Siegwart, K.O. Arras, S. Bourabdallah, D. Burnier, G. Froidevaux, X. Greppin, B. Jensen, A. Lorotte, L. Mayor, M. Meisser, R. Phillipsen, R. Piguet, G. Ramel, G. Terrien, and N. Tomatis, "Robox at expo.02: A large-scale installation of personal robots," in *In Proceedings of IEEE int. Conf. on Robotics and Automation(ICRA)*, 2003, pp. 203–222.

[13] P. Biber and T. Duckett, "Dynamic maps for long-term operation of mobile service robots," *Robotic Science and Systems*, pp. 17–24, 2005.

[14] A Milstein, "Dynamic maps in monte carlo localization," *Lecture Notes in Computer Science*, pp. 1–12, 2005.

[15] E.W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, pp. 269–271, 1959.