

## Research Article

# An Adaptive Memory Model for Long-Term Navigation of Autonomous Mobile Robots

**M. Hentschel and B. Wagner**

*Department of Real Time Systems, Institute for Systems Engineering, Leibniz Universität Hannover, 30167 Hannover, Germany*

Correspondence should be addressed to M. Hentschel, hentschel@rts.uni-hannover.de

Received 14 July 2011; Accepted 9 October 2011

Academic Editor: Jorge Manuel Dias

Copyright © 2011 M. Hentschel and B. Wagner. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper introduces an environmental representation for autonomous mobile robots that continuously adapts over time. The presented approach is inspired by human memory information processing and stores the current as well as past knowledge of the environment. In this paper, the memory model is applied to time-variant information about obstacles and driveable routes in the workspace of the autonomous robot and used for solving the navigation cycle of the robot. This includes localization and path planning as well as vehicle control. The presented approach is evaluated in a real-world experiment within changing indoor environment. The results show that the environmental representation is stable, improves its quality over time, and adapts to changes.

## 1. Introduction

The ability to navigate in the environment is one of the most important requirements for an autonomous mobile robot. In general, this navigation task can be defined as the combination of three basic competences: localization, path planning, and vehicle control. Localization denotes the robot's ability to determine its own position and orientation (referred to as *pose*) within a global reference frame. Path planning defines the calculation of an adequate sequence of motion commands to reach a desired destination from the current robot position. Due to its planning component, path planning is typically executed before motion. The planned path is followed by the robot using motion control and reactive obstacle avoidance. In case an obstacle is reactively unavoidable, global path replanning is performed.

In order to solve the navigation task, previous knowledge about the environment is required. This knowledge includes landmark features which may be used for localization, geometric and semantic information about routes which may be applied to path planning, and obstacle configurations which must be circumvented safely. Within the robotic community, the environmental knowledge is typically given in advance [1] or built up from scratch before operation by using SLAM approaches [2].

In this case, it is supposed that the world can be assumed as static or that the robot is operated only for a limited period of time. However, future service robots are required to run autonomously from several weeks up to years. During this long-term operation, these robots are expected to share their workspace with people and to interact with humans as well as with manually and autonomously operated vehicles. Typically, these environments change over time and cannot be assumed as static any longer.

Changes in the environment may appear gradually or in an abrupt way when they occur outside the sensory range of the robot. In general, we differentiate three types of objects: *dynamic*, *semistatic*, and *static* objects. Objects like cars or people moving with a certain velocity in a defined direction are called dynamic. Within the sensory range of the robot, the velocity and movement direction can be estimated by considering subsequent sensory measurements. Objects that change their pose or physical dimensions without a direct movement are called semistatic, such as growing trees and parking cars as well as changes that occur outside the sensory range of the robot. On the other side, objects which are invariant to changes are denoted as static. Furthermore, changes may not be permanent: a door may have been opened, a package may have been left in the corridor for

a while, and so forth. In general, it is unknown to the autonomous robot when, where, and for how long these changes will occur. For a secure and safe long-term operation in these environments, mobile robots will have to adapt to these changes. Moreover, in giving robots the ability to store the current state and to memorize the past appearances of the environment as well, they will have the ability to learn from the past. These demands require new concepts for the environmental representation of mobile systems.

In this paper, an adaptive environmental model for long-term navigation of autonomous mobile robots is presented. Following the concept of human memory, this representation consists of three basic stores: sensory memory, short-term memory, and long-term memory. The sensory memory contains raw data from the sensors while the short-term memory is used to store environmental data that is of vital importance for the robot. Within the short-term memory, relevant data is identified and then transferred to the long-term memory. By using temporal parallel stores, the presented memory approach allows a spatial and temporal modelling of time-variant environmental data. In this paper, the memory model is applied to obstacles and routes in the environment and used for object tracking, localization, and path planning as well as obstacle avoidance. Therefore, the adaptive memory model forms the basis for solving the navigation task of the autonomous robot during long-term operation.

The rest of the paper is structured as follows. Section 2 discusses previous work on navigation approaches in changing environments. Section 3 introduces the adaptive memory model this work is based on. Section 4 presents the update of the obstacle representation and Section 5 the update of the route data in the memory model. In Section 6 the data included in the memory model is used for autonomous robot navigation. Real-world experimental results in indoor environments are given in Section 7. Finally, the conclusion and discussion for future work are presented in Section 8.

## 2. Related Work

Most of the existing navigation approaches assume the world to be static. To deal with changing environments, some approaches suppose that a specific part of the sensory data is static and invariant to changes over time. For example Soika et al. [3] and Wulf et al. [4] extract ceiling features from a 3D point cloud. Within the scan points, these ceiling features are assumed to be time invariant and robust against occlusion by moving objects. The extracted features are used as landmark measurements for localization in industrial halls based on an a priori given map.

Other approaches attempt to explicitly discriminate dynamic from static measurements and detect and filter out moving objects. For example, Fox et al. [5] use an entropy filter to identify measurements caused by dynamic objects. Thrun et al. [6] developed for the well-known Minerva tour guide robot a distance filter to separate sensor readings corresponding to known objects from readings caused by dynamic obstacles. Therefore a preinstalled map of all known objects was used. In SLAM approaches, Wang et al. [7] employ a feature-based heuristic to identify and filter out

dynamic objects in range measurements. Hähnel et al. [8] use a probabilistic method for tracking people and filter out the corresponding measurements to improve the map building process. Although these filtering approaches have been proven to be robust in highly dynamic environments, they are unable to detect semistatic objects.

In recent years, few authors started to research in long-term localization and explicitly model the changes in the environment. Biber and Duckett [9] propose a spatio-temporal map where the environment is represented at multiple timescales simultaneously. By replenishing parts of the map with new sensor measurements by a timescale-specific learning rate, the dynamic map adapts continuously over time. The representation is used for localization by selecting the timescale of the map which best fits to the current sensor data. Stachniss and Burgard [10] introduced an approach of modelling typical configurations of dynamic environments like open and closed doors. In areas where changes are detected, this approach creates local grid-based maps (patch maps) and estimates for each sub-map clusters of possible configurations in the environment. These patch maps are integrated into a Monte-Carlo localization. This work is extended by Meyer-Delios et al. [11] by using local temporary maps to improve the localization in semistatic environments like a parking lot. Dayoub and Duckett [12] presented an approach for long-term topological localization based on omnidirectional camera vision. Local features are extracted from panoramic images to represent the appearance of a node in the topological map. By adopting concepts of short-term and long-term memory, the presented approach updates the group of feature points for the reference image of a particular place.

However, all existing approaches for long-term operation of mobile robots focus on the localization part. So far, the complete cycle of autonomous navigation, consisting of localization, path planning, and vehicle control is not considered.

## 3. The Adaptive Memory Model

In this paper all knowledge about the environment is supposed to be a function of time. Assuming a planar world, the configuration space of the environment in spatial as well as in temporal domain tends in long-term operation towards infinity. To represent present as well as past knowledge about the environment of an autonomous mobile robot with limited memory capacities, an adaptive memory model is introduced (see Figure 1). The memory model is based on the multistore model of human memory proposed in 1968 by Atkinson and Shiffrin [13]. This model, which forms the basis of modern memory theories, divides human memory into the three basic stores: *sensory memory* (SM), *short-term memory* (STM), and *long-term memory* (LTM).

According to Atkinson and Shiffrin, the sensory memory contains all raw data perceived by the senses. While encoding the data in SM, elementary identification processes are performed. As the amount of data is huge, the stored data decays in SM after a period of approximately 2 seconds. The short-term or working memory forms the basis of

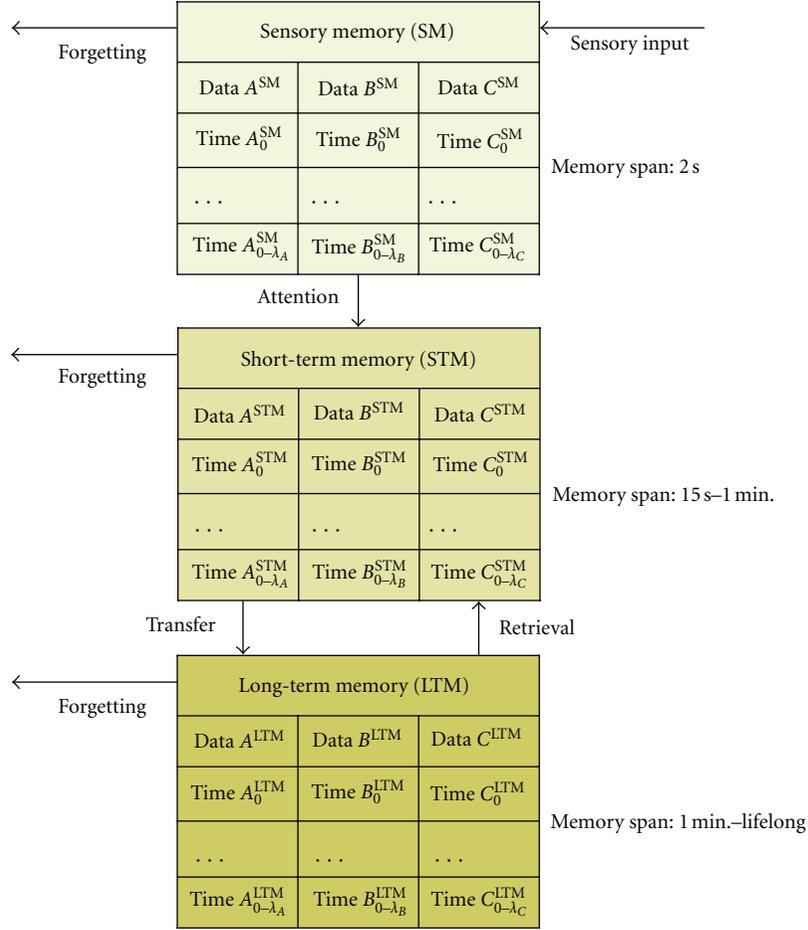


FIGURE 1: Proposed memory model.

mental information processing. Selective attention mechanisms determine which data is moved from the sensory memory to STM. As the capacity of the short-term memory is limited, further abstraction is performed while storing the data. The data in STM can be recalled for a period of several seconds to a minute before being forgotten. Through the process of rehearsal, data in STM can be transferred to the long-term memory to be retained for longer periods of time. During the storage process in LTM, further abstraction and interpretation as well as combination with encoded data are performed. In return, the knowledge stored in LTM affects the perception and influences the data attended to in the environment. In general, when new data moves from SM via STM to LTM, the amount of data is condensed with previous data and the level of abstraction arises.

In this paper we are applying this human memory concept to autonomous mobile robots. For this, we make two assumptions. Firstly, each store of the memory (SM, STM, and LTM) is divided into substores  $\zeta$  to represent different kinds of data (e.g., obstacle and route data). Secondly, each substore  $\zeta$  is divided in temporal domain into  $\lambda$  slots of equal length  $\Delta t$ . Within the time  $\Delta t$  the data is assumed to be valid. For this, each substore is able to memorize a period of time of  $\lambda \cdot \Delta t$ . The temporal representation is assumed to be a circular

buffer, overwriting the first entries for an operation time of more than  $\lambda \cdot \Delta t$ . With a current time given by  $t$ , the index  $\tau$  of the current temporal slot in substore  $\zeta$  is computed by applying the modulus operation as follows:

$$\tau = \frac{(t - t_s)}{\Delta t} \bmod \lambda, \quad (1)$$

where  $t_s$  denotes the starting time of operation. The number of  $\lambda$  temporal slots per store is predefined and depends on the kind of data stored. In general, the time  $\Delta t$  for each temporal slot increases from SM over STM to LTM resulting in an increasing memory span of  $\lambda \cdot \Delta t$ .

In the following sections, we will present how we integrate obstacles as well as route knowledge in this memory concept and use the memorized information to improve robot navigation in changing environments (see Figure 2 for an overview).

## 4. Obstacle Representation

**4.1. Sensory Memory.** For environmental perception, 2d and 3d range sensors are supposed. Using only 2d approaches for navigation, the relevant 3d sensor data is reduced to the 2d plane by applying the technique of *virtual 2d scans* [14]. Each

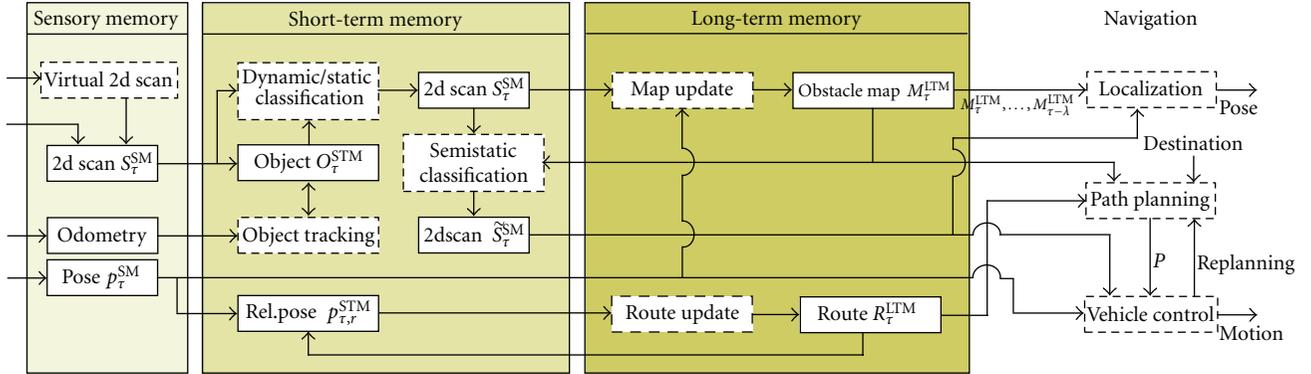


FIGURE 2: Dataflow of the long-term navigation approach. Solid white rectangles denote subslots in the memory model, and dotted rectangles represent algorithms for data processing.

single 2d sensor measurement  $s_r^{SM}$ , named scan point below, is defined as follows:

$$s_r^{SM} = (\alpha, r, d)^T \quad \text{with } \alpha, r \in \mathbb{R}, d \in D, \quad (2)$$

where  $\alpha$  denotes the polar angle and  $r$  the range of the measurement in the robot-centred polar coordinate system  $CS_R$ . The element  $d$  represents an estimate of the dynamic class of the scan point, being an element of the following possible set:

$$D = \{\text{dynamic, semistatic, static, unknown}\}. \quad (3)$$

Following the definition in Section 1, *dynamic* indicates that the scan point belongs to a moving object whereas *semistatic* represents an object which changes without moving and *static* indicates objects time invariant to changes. *Unknown* denotes that there is no dynamics classification available. Since common range sensors are unable to classify the dynamics of the scan points,  $d$  is initially set to *unknown* for  $s_r^{SM}$ .

Since the polar angle  $\alpha$  and the range  $r$  of the scan point are measured with a real sensor, they are limited in their precision. For modelling the errors, we assume for each  $s_r^{SM}$  a normal distributed error with the following covariance:

$$\sum_{\tau, s}^{SM} = \begin{pmatrix} \sigma_\alpha^2 & \sigma_{r\alpha} \\ \sigma_{\alpha r} & \sigma_r^2 \end{pmatrix}. \quad (4)$$

Note that we are neglecting systematic measurement errors and model only the randomized errors of  $\alpha$  and  $r$ . For further simplification, we suppose the dynamic classification  $d$  to be “correct.”

Finally, a full 2d scan representing the local environment around the robot is defined as follows:

$$S_r^{SM} = \left\{ s_{r,i}^{SM}, \sum_{\tau, s, i}^{SM} \right\}_{i=1 \dots n_r^{SM}}. \quad (5)$$

The scan  $S_r^{SM}$  consists of  $n_r^{SM}$  scan points  $s_{r,i}^{SM}$  and the associated covariance matrices  $\sum_{\tau, s, i}^{SM}$ . Each 2d scan is related to a discrete time step  $\tau$ . In case where multiple sensors are

used, an additional transformation applies transforming the sensor data into the robot coordinate system  $CS_R$ .

Assuming a ground-based vehicle, the pose of the robot is defined at every time step  $\tau$  as follows:

$$p_r^{SM} = (x, y, \varphi)^T \quad \text{with } x, y, \varphi \in \mathbb{R}, \quad (6)$$

where  $x$  and  $y$  represent the position and  $\varphi$  represents the orientation of the robot (to be more precise, the pose of the robot coordinate system  $CS_R$ ). The pose  $p_r^{SM}$  is given in the world-centered Cartesian coordinate system  $CS_W$  and is the result of the localization approach (see Section 6.1). Depending on the input data and the localization approach, the pose is estimated with a certain amount of uncertainty.

Assuming a normal distributed error in position as well as orientation, the related covariance matrix is defined at every time step  $\tau$  as follows:

$$\sum_{\tau, p}^{SM} = \begin{pmatrix} \sigma_x^2 & \sigma_{yx} & \sigma_{\varphi x} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{\varphi y} \\ \sigma_{x\varphi} & \sigma_{y\varphi} & \sigma_\varphi^2 \end{pmatrix}. \quad (7)$$

**4.2. Short-Term Memory.** In short-term memory, an abstract object representation is created from the 2d scan  $S_r^{SM}$ . For this, the scan is segmented and scan points which are related together in spatial domain are grouped by a bounding box  $o_r^{STM}$ . Each bounding box is defined as follows:

$$o_r^{STM} = (x, y, \varphi, l, w, v, d)^T, \quad (8)$$

where  $x$  and  $y$  represent the position and  $\varphi$  represents the orientation of the centre point of the bounding box in Cartesian coordinates given in the robot coordinate system  $CS_R$ .  $l$  and  $w$  are the length and width of the box with  $l$  pointing towards  $\varphi$ . The absolute velocity in direction  $\varphi$  is denoted by  $v$  and the dynamic classification of the object defined by  $d \in D$ . Within each 2d scan  $S_r^{SM}$ ,  $n_r^{STM}$  bounding box representations are included and given by the following set  $O_i^{STM}$ :

$$O_r^{STM} = \{o_{\tau, j}^{STM}\}_{j=1 \dots n_r^{STM}}. \quad (9)$$

The bounding box representation is used to estimate the movement of objects in the environment. Therefore, the poses of all objects belonging to  $O_t^{\text{STM}}$  are tracked in time. For this, a common Kalman-Filter approach is applied as described in [7]. By predicting the object motion according to a linear motion model and using the nearest neighbor criteria, the tracking approach associates to each object  $o_{\tau,j}^{\text{STM}}$  at time  $\tau$  a corresponding object  $o_{\tau-1,k}^{\text{STM}}$  at the previous time  $\tau - 1$ . Between these time steps, the ego motion of the robot is measured and compensated using wheel odometry. From the change in object position between subsequent time steps, the absolute velocity  $v$  as well as the moving direction  $\varphi$  at the current time step  $\tau$  is estimated. Next, the estimated velocity is used to classify the dynamics of the tracked objects following the straightforward metric:

$$d_j = \begin{cases} \text{dynamic} & \text{if } v_j \geq v_{\text{thres}}, \\ \text{static} & \text{if } v_j < v_{\text{thres}}, \\ \text{unknown} & \text{else object not tracked.} \end{cases} \quad (10)$$

In case the velocity  $v_j$  is equal to or exceeds the threshold  $v_{\text{thres}}$ , the object is assumed as being dynamic. Otherwise, the object is classified as static. If no data association between the current and the previous objects is possible, the object's dynamics is assumed as unknown.

Next, the tracked objects are used to classify the dynamics of the scan points in short-term memory. For this, each scan point  $s_{\tau,i}^{\text{SM}} \in S_{\tau}^{\text{SM}}$  is checked if an object  $o_{\tau,j}^{\text{STM}} \in O_{\tau}^{\text{STM}}$  exists that includes the scan point. In case  $s_{\tau,i}^{\text{SM}}$  is included in an object, the dynamics of the object is used. Otherwise, the dynamics is assumed as unknown:

$$d_i = \begin{cases} d_j & \text{if } s_{\tau,i}^{\text{SM}} \text{ is inside } o_{\tau,j}^{\text{STM}}, \\ \text{unknown} & \text{else.} \end{cases} \quad (11)$$

In short-term memory, the result is stored in a 2d scan  $S_{\tau}^{\text{STM}}$  which corresponds to the scan  $S_{\tau}^{\text{SM}}$  with the dynamic of each scan point classified by the tracking approach.

**4.3. Long-Term Memory.** The obstacle representation  $M_{\tau}^{\text{LTM}}$  in long-term memory is assumed as an occupancy grid map. The occupancy grid is a 2d tessellation of space into discrete cells, where each cell stores a probabilistic estimate of its state. Following the work by Elfes [15], the state  $s(C)$  associated with each cell  $C \in M_{\tau}^{\text{LTM}}$  is defined as a discrete random variable with the two states *occupied* (OCC) and *empty* (EMP). As both states are exclusive, and exhaustive the probability follows the rule  $P[s(C) = \text{OCC}] + P[s(C) = \text{EMP}] = 1$ .

The update step of the long-term memory  $M_{\tau}^{\text{LTM}}$  is based upon the scan representation  $S_{\tau}^{\text{STM}}$  in short-term memory and the current robot pose  $p_{\tau}^{\text{SM}}$ . For deciding which scan points  $s_{\tau,i}^{\text{STM}} \in S_{\tau}^{\text{STM}}$  to be transferred from short-term to long-term memory, an update rule inspired by the work of Biber and Duckett [9] is used. Depending on an update rate  $0 \leq u \leq 1$ , the scan points for the update process of LTM are selected randomly by performing the following steps for each new 2d scan in STM:

- (i) choose all  $n_{\tau,\text{stat}}^{\text{STM}}$  scan points with  $d = \text{static}$  in the current 2d scan  $S_{\tau}^{\text{STM}}$ ;
- (ii) select  $u \cdot n_{\tau,\text{stat}}^{\text{STM}}$  randomly chosen scan points from the static scan points in  $S_{\tau}^{\text{STM}}$ ;
- (iii) Add the  $u \cdot n_{\tau,\text{stat}}^{\text{STM}}$  scan points to the temporary set  $S'$ .

For each new 2d scan in STM, the set  $S'$  is initially defined as  $S' = \emptyset$ .

As defined in Section 4.1, the scan is originated in robot coordinates  $\text{CS}_R$ . For updating the obstacle representation  $M_{\tau}^{\text{LTM}}$  in world coordinates  $\text{CS}_W$ , a coordinate transformation  $F(s'_{\tau,i}, p_{\tau}^{\text{SM}})$  for the coordinates as well as the uncertainties of each scan point applies. With the pose  $p_{\tau}^{\text{SM}}$  of the robot, the scan point  $s'_{\tau,i}$  in world coordinates is calculated as follows:

$$s'_{\tau,i} = F(s'_{\tau,i}, p_{\tau}^{\text{SM}}) = \begin{pmatrix} x_{\tau} + r'_{\tau,i} \cdot \cos(\varphi_{\tau} + \alpha'_{\tau,i}) \\ y_{\tau} + r'_{\tau,i} \cdot \sin(\varphi_{\tau} + \alpha'_{\tau,i}) \end{pmatrix}. \quad (12)$$

According to (4) and (7), the scan point as well as the pose of the robot is affected by uncertainty. Therefore, the resulting covariance matrix of the scan point in world coordinates is calculated as follows:

$$\sum''_{\tau,s,i} = \nabla F(s'_{\tau,i}) \sum'_{\tau,s,i} \nabla F(s'_{\tau,i})^T. \quad (13)$$

Here,  $\nabla F(s'_{\tau,i})$  is the Jacobian matrix linearized at the position of the transformed scan point and  $\sum'_{\tau,s,i}$  defines the combined covariance matrix of the pose and the scan points:

$$\sum'_{\tau,s,i} = \begin{pmatrix} \sum_{\tau,p} & 0 \\ 0 & \sum_{\tau,s,i} \end{pmatrix}. \quad (14)$$

With the scan point  $s'_{\tau,i}$  and the covariance matrix  $\sum''_{\tau,s,i}$  in global coordinates, the resulting probability distribution for each scan point is defined as follows:

$$p''_{\tau,i}(x) = \frac{1}{\sqrt{(2\pi)^2 \det(\sum''_{\tau,s,i})}} \times \exp\left(-\frac{1}{2}(\vec{x} - s'_{\tau,i})^T \sum''_{\tau,s,i}^{-1} (\vec{x} - s'_{\tau,i})\right). \quad (15)$$

This probability distribution is used for updating the obstacle representation as described by Elfes [15] using the Bayesian reasoning.

**4.4. Semistatic Classification.** Following the update step of the long-term obstacle representation,  $M_{\tau}^{\text{LTM}}$  is used for classifying the semistatic obstacles in short-term memory. Therefore, for each scan point  $s_{\tau,i}^{\text{STM}}$  in short-term memory, the obstacle representation  $M_{\tau}^{\text{LTM}}$  is checked if an obstacle is within the uncertainty of the scan point. In case no equivalent to the scan point is found in the map, the dynamics is set to semistatic. The classified scan point is stored within the set  $S_{\tau}^{\text{STM}}$  in short-term memory.

## 5. Route Representation

The routes which may be used for autonomous navigation are defined by a set  $R$  of  $n$  possible routes:

$$R = \{r_i\}_{i=1\dots n}. \quad (16)$$

Each single route  $r_i$  consists of a directed set of  $m_i$  waypoints  $w_{i,j}$ :

$$r_i = \{w_{i,j}\}_{j=1\dots m_i} = \left\{ \left( x_{i,j}, y_{i,j}, v_{i,j}, l_{i,j}, t_{i,j}, \Delta t_{i,j} \right)^T \right\}_{j=1\dots m_i}, \quad (17)$$

where  $(x_{i,j}, y_{i,j})$  is the waypoint location in global Cartesian coordinates and  $v_{i,j}$  defines the maximum velocity for approaching the waypoint. The width of the route corridor between two subsequent waypoints  $w_{i,j-1}$  and  $w_{i,j}$  is defined by  $l_{i,j}$  which is the perpendicular distance from the route centre line to the boundary edge.  $\Delta t_{i,j}$  denotes the required travel time between the subsequent waypoints  $w_{i,j-1}$  and  $w_{i,j}$ , and  $t_{i,j}$  represents the last update time of any parameter of this route.

The update process of the route representation is based on our former work. For more details, please refer to [16].

**5.1. Short-Term Memory.** In short-term memory, the route segment which the robot is currently on is estimated. Therefore the pose  $p_{\tau,r}^{\text{STM}}$  of the robot relative to the route representation  $R_{\tau}^{\text{LTM}}$  in long-term memory is computed. This is done by calculating the pose relative to all available routes in  $R_{\tau}^{\text{LTM}}$  and selecting the pose where the lateral distance to the route is minimal. In case the relative pose is within the boundaries of the route,  $p_{\tau,r}^{\text{STM}}$  is stored within the short-term memory.

**5.2. Long-Term Memory.** The relative pose is used for updating the route representation  $R_{\tau}^{\text{LTM}}$  in long-term memory. Here we differentiate two cases. If the pose  $p_{\tau,r}^{\text{STM}}$  is within a given route corridor, the waypoints of the route are updated. Next to the coordinates  $(x_{i,j}, y_{i,j})$  of the waypoints, the travel time  $\Delta t_{i,j}$  and the time  $t_{i,j}$  are updated once the route segment is passed by the robot. In case the pose  $p_{\tau,r}^{\text{STM}}$  is outside the existing route representation, a new route is added to  $R_{\tau}^{\text{LTM}}$ . For this, a new waypoint is added to the representation whenever the distance  $\Delta d$  or the angular difference  $\Delta\phi$  to the last waypoint exceeds a predefined threshold. In addition, in every update step of the long-term memory, the age of the route knowledge is considered. Whenever a route representation in long-term memory is older than a certain threshold without being updated by the robot, the route is removed from the long-term memory and forgotten.

## 6. Long-Term Navigation

The memory model presented in this paper is used for solving the navigation task of the robot. Therefore, the spatio-temporal representation is integrated into localization, path planning, and vehicle control of the autonomous system.

**6.1. Localization.** For localization, a Monte-Carlo localization is used as presented in [17]. Sensory input of the localization is the 2d scan  $\hat{\mathcal{S}}_{\tau}^{\text{STM}}$  of all scan points at time  $\tau$ . Within this scan, only the scan points which are classified as *static* are regarded for localization. As reference map for localization, the static obstacle map  $M_{\tau_i}^{\text{LTM}}$  in long-term memory at time  $\tau_i$  is used. To maintain  $M_{\tau_i}^{\text{LTM}}$ , all obstacle maps in LTM are searched for the map that fits best to the current sensory input  $\hat{\mathcal{S}}_{\tau}^{\text{STM}}$ . This is achieved by calculating for each obstacle map in LTM the expected range measurements and selecting that map whose overall mean square error between the measured and the expected range measurements is minimal.

**6.2. Path Planning.** For path planning, an optimal path from the current robot pose  $p$  to the given destination is searched. Considering the route data stored in LTM, the actual route representation  $R_{\tau}^{\text{LTM}}$  is used for path planning. Within the representation of all admissible ways at time  $\tau$ , path planning is performed using the well-known  $A^*$  search algorithm. This algorithm computes a least-cost path from the starting pose to the given destination minimizing the overall travel time along the way. Estimating the travel costs along each route, the travel times stored in LTM and a heuristic estimate of the remaining costs to the destination are used. The heuristic estimate is based on the travel time from the straight-line distance divided by the maximum robot velocity.

The result path of the planning algorithm is a set  $P$  of  $m$  waypoints  $w$ :

$$P = \{w_i\}_{i=1\dots m}. \quad (18)$$

These waypoints define the base trajectory of the resultant path as well as the corridor width along the path.

**6.3. Vehicle Control.** Following the preplanned path  $P$  is achieved by a hybrid feedback controller introduced by Hentschel et al. in [18]. This controller enables precise tracking of the planned path as well as obstacle avoidance. The obstacle avoidance consists of two parts, reactive obstacle avoidance based on the current sensory data  $\hat{\mathcal{S}}_{\tau}^{\text{STM}}$  and global path replanning. Reactive obstacle avoidance is achieved by adjusting the lateral offset of the robot with respect to the base trajectory of the path. For each path segment, the lateral displacement is upper bounded by the lateral boundary of the path corridor  $l$ . Next to obstacle avoidance, the vehicle speed is reduced in dependence of the distance to the perceived obstacles. In addition, global path replanning is performed to circumvent obstacles which cannot be avoided reactively. For global path replanning, the current obstacle map  $M_{\tau}^{\text{LTM}}$  is considered. Within the known obstacle configuration, the optimal combination of motion commands is searched which circumvents the obstacles and returns as fast as possible behind the obstacle to the path  $P$ . This motion search is done as well in obstacle map by  $A^*$  algorithm.

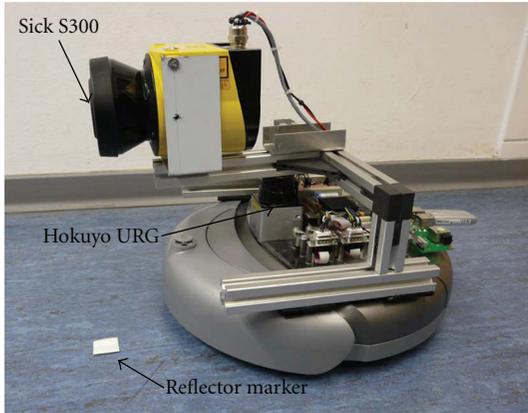


FIGURE 3: Roomba vacuum cleaner used in the experimental results.

## 7. Experimental Results

**7.1. Experimental Setup.** To demonstrate the applicability of the presented memory model for long-term navigation of autonomous mobile robots, a real-world indoor experiment is performed. For this, an *iRobot Roomba SE* vacuum cleaner (see Figure 3) is used as robotic platform. Odometry measurements of the differential drive are received and movement commands are sent at a rate of 10 Hz via the open *Roomba Serial Command Interface (SCI)*. For environmental perception, the robot is equipped with a *Hokuyo URG-04LX 2d* laser range scanner with a field of view of  $240^\circ$  at a maximum range of 4 m and an update rate of 10 Hz. All required algorithms for navigation as well as data acquisition are computed in real-time using an *AMD Geode LX800* embedded pc with *Linux/Xenomai* real time operating system on board of the robot. For evaluation purpose all sensory data is logged as well.

The experiment is conducted in an apartment style environment with the dimensions of 8 m by 10 m. The workspace includes typical apartment furniture like cupboards, tables, sofas, and flower tubs. For the robot, initial knowledge about the environment is given in advance (see Figure 4). This knowledge includes the position of the walls within the environment as well as one desired route.

Obstacles in the environment are unknown in advance and must be learned automatically by the adaptive memory model.

In this environment, the robot is operated four weeks (28 days) fully autonomously with one run per day. In the first week, the environment is completely static while in the second week various changes are inserted. This includes new obstacles as well as changing obstacle configurations. For the third week, two persons are entering the environment, affecting and traversing the path of the robot. In the last week of the experiment, the environment remains static, but on day 22 parts of the path are fully blocked by an obstacle.

In this experiment, the memory model is parameterized as follows:  $\Delta t^{\text{SM}} = 0.1 \text{ s}$ ,  $\lambda^{\text{SM}} = 10$ ,  $\Delta t^{\text{STM}} = 0.1 \text{ s}$ ,  $\lambda^{\text{STM}} = 100$ ,  $\Delta t^{\text{LTM}} = 8640 \text{ s}$ , and  $\lambda^{\text{LTM}} = 10$ . For the grid size of the obstacle map in long-term memory, 50 mm by 50 mm

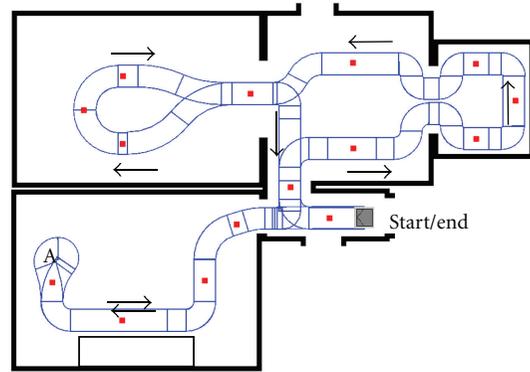


FIGURE 4: Initial environmental knowledge given to the robot. The black line indicates the landmarks for localization, the blue represents the route for path planning, and the red squares indicate the position of the reflector markers for ground truth. Arrows represent the direction of motion.

is chosen. The update rate of the long-term memory is set to  $u = 0.05$ . By this means, 5 percent of the static 2d scan points in short-term memory are used to update the LTM. During the experiment the top speed of the robot is 0.2 m/s. The threshold for detecting dynamic obstacles is defined as  $v_{\text{thres}} = 0.3 \text{ m/s}$ .

**7.2. Ground Truth.** To evaluate the localization result over time, ground truth data for the robot pose is required. As we are considering the full navigation cycle of an autonomous robot, ground truth data must be available during motion while the path of the robot may vary over time.

For this, 15 artificial reflector markers with a squared size of 25 mm by 25 mm are used. The markers are located at regular distances on the centre line of the ideal path and are surveyed manually in their location. During motion, the reflectors are detected by an additional *Sick S300 2d* laser scanner scanning towards ground in front of the robot (see Figure 3). Within a lateral distance of 0.5 m, the reflector position is measured in robot coordinates. With the pose of the robot localization, the global reflector position is calculated and compared with the ground truth reflector positions for evaluation.

**7.3. Results.** During this experiment, the robot traveled a total distance of 1225 m fully autonomously and acquired 37800 2d scans. Starting with the initial representation in Figure 4, new obstacles are learned and added to the long-term representation. Figure 5 represents the LTM after day seven.

Within the short-term memory, the scan points which do not belong to dynamic objects and are not included in the long-term map are classified as semistatic. As the obstacle knowledge condenses over time, the average number of semistatic scan points included in the 2d scan drops from 23% on day one to 3% on day seven (see Figure 6). During the second week, the obstacle configuration is changed every day. As a consequence, the percentage of semistatic points

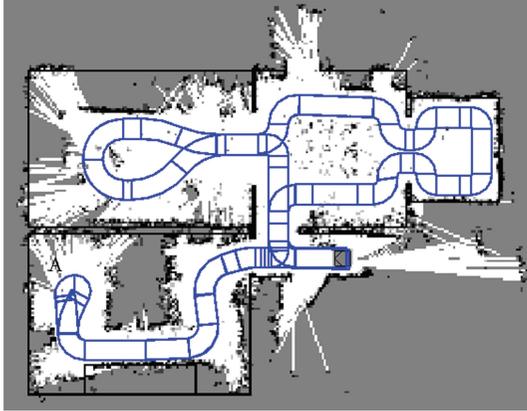


FIGURE 5: Environmental representation after day seven. The initial knowledge is condensed by the perceived obstacles.

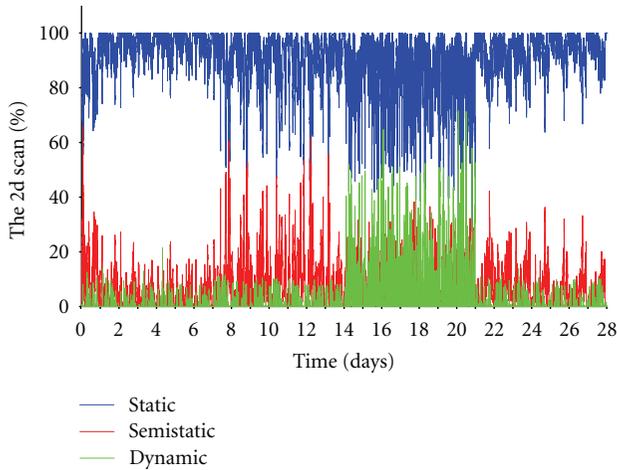


FIGURE 6: Dynamic classification of the 2d scans using the adaptive memory model.

included in the 2d scan increases to an average of 9.5%. Due to the dynamic objects, the number of dynamic scan points rises from an average of 0.5% to 9.2% in the third week. As the environment remains static in the last week, the percentage of dynamic and semistatic scan points is fairly similar to the end of week one.

By the cumulative knowledge about the environment in the LTM, the mean error in position with respect to ground truth decreases from 0.14 m on day one to 0.09 m on day seven (see Figure 7). As there are no equivalent obstacle representations in LTM, the changing obstacle configurations in the second week cause an increasing position error. Especially at day nine the maximum position error increases up to 0.48 m. On this day, a large obstacle was inserted at point A (see Figure 5) which occluded most of the known obstacles. During the experiment, the average position error was 0.11 m.

To demonstrate the benefits of our presented approach, the position errors are compared to a common static Monte-Carlo localization approach. This is done via post-processing

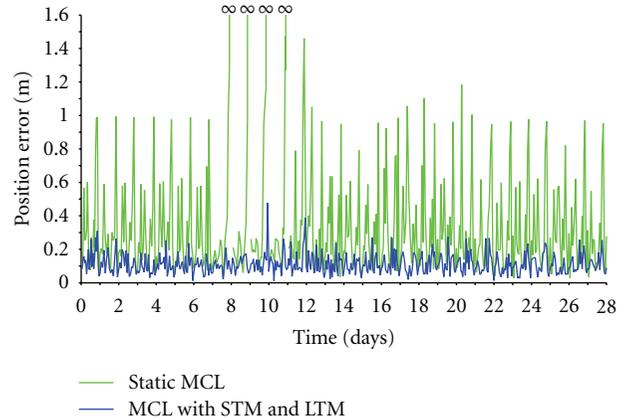


FIGURE 7: Variation of the position error over time using the adaptive localization and in comparison to a static localization.

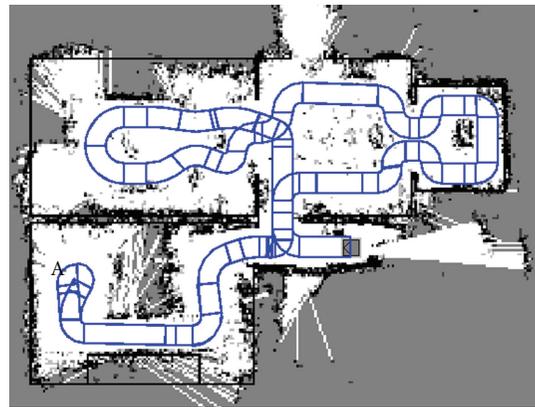


FIGURE 8: Environmental representation after day 22. The route representation is adapted to avoid the obstacle blocking the initial path.

using the logged sensory data and the initial knowledge from Figure 4 as the environmental representation.

Using the static MCL, the position error is noticed to be higher than when using our adaptive approach. Furthermore, on the days 8, 9, 10, and 11, the static localization is unable to calculate a pose estimate due to the obstacle configuration which occluded most of the walls in the environment.

During the day 22, the route is blocked by an obstacle (see Figure 8). By using global path replanning and the obstacle representation in LTM, the robot circumvents the obstacle and follows the initial route, adapting the route representation in LTM.

## 8. Conclusion

In this paper, an adaptive memory model for long-term navigation of autonomous mobile robots in changing environment is presented. The proposed memory model consists of the three basic stores: sensory memory, short-term memory, and long-term memory. By using temporal

parallel stores, the memory approach allows a spatial and temporal representation of time-variant environmental data. In this paper, obstacles as well as route data are integrated in the memory. By analyzing the stored information over time, the knowledge about the environment is condensed and used for solving the navigation task of autonomous mobile robots.

Future efforts will concentrate on intensifying the experiments regarding long-term operation and to extend our activities to outdoor environment. In addition, more of the environmental information, for example, object representation, should be stored in the memory model.

## References

- [1] C. Urmson, J. Anhalt, D. Bagnell et al., "Autonomous driving in urban environments: boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 1556–4959, 2008.
- [2] J. Sprickerhof, A. Nüchter, K. Lingemann, and J. Hertzberg, "An explicit loop closing technique for 6D SLAM," in *Proceedings of the 4th European Conference on Mobile Robots (ECMR '09)*, Mlini/Dubrovnik, Croatia, September 2009.
- [3] M. Soika, S. Pook, and W. Feiten, "Autonomous robot navigation with 3D laser measurements," in *International Symposium on Robotics*, 2006.
- [4] O. Wulf, D. Lecking, and B. Wagner, "Robust self-localization in industrial environments based on 3D ceiling structures," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pp. 1530–1534, October 2006.
- [5] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *Journal of Artificial Intelligence Research*, vol. 11, pp. 391–427, 1999.
- [6] S. Thrun, M. Beetz, M. Bennewitz et al., "Probabilistic algorithms and the interactive museum tour-guide robot Minerva," *International Journal of Robotics Research*, vol. 19, no. 11, pp. 972–999, 2000.
- [7] C. C. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '03)*, pp. 842–849, September 2003.
- [8] D. Hähnel, D. Schulz, and W. Burgard, "Map building with mobile robots in populated environments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 496–501, October 2002.
- [9] P. Biber and T. Duckett, "Dynamic maps for long-term operation of mobile service robots," in *Proceedings of the Robotics: Science and Systems (RSS '05)*, pp. 17–24, Cambridge, Mass, USA, June 2005.
- [10] C. Stachniss and W. Burgard, "Mobile robot mapping and localization in non-static environments," in *Proceedings of the 20th National Conference on Artificial Intelligence and the 17th Innovative Applications of Artificial Intelligence Conference (AAAI '05/IAAI '05)*, pp. 1324–1329, July 2005.
- [11] D. Meyer-Delios, J. Hess, G. Grisetti, and W. Burgard, "Temporary maps for robust localization in semi-static environments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 5750–5755, Taipei, Taiwan, 2010.
- [12] F. Dayoub and T. Duckett, "An adaptive appearance-based map for long-term topological localization of mobile robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '08)*, pp. 3364–3369, September 2008.
- [13] R. C. Atkinson and R. M. Shiffrin, "Human memory: a proposed system and its control processes," in *The Psychology of Learning and Motivation*, K. W. Spence and J. T. Spence, Eds., vol. 2, pp. 89–195, Academic Press, New York, NY, USA, 1968.
- [14] O. Wulf, C. Brenneke, and B. Wagner, "Colored 2D maps for robot navigation with 3D sensor data," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '04)*, pp. 2991–2996, October 2004.
- [15] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.
- [16] M. Hentschel and B. Wagner, "Adaptive path planning for long-term navigation of autonomous mobile robots," in *Proceedings of the 4th European Conference on Mobile Robots (ECMR '09)*, Mlini/Dubrovnik, Croatia, September 2009.
- [17] O. Wulf, M. Khallaf-Allah, and B. Wagner, "Using 3D data for Monte Carlo localization in complex indoor environments," in *Proceedings of the 2nd Bi-Annual European Conference on Mobile Robots (ECMR '05)*, pp. 170–175, Ancona, Italy, 2005.
- [18] M. Hentschel, O. Wulf, and B. Wagner, "A hybrid feedback controller for car-like robots—combining reactive obstacle avoidance and global replanning," *Integrated Computer-Aided Engineering*, vol. 14, no. 1, pp. 3–14, 2007.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

