Toward an Object-Based Semantic Memory for Long-Term Operation of Mobile Service Robots

Feras Dayoub, Tom Duckett and Grzegorz Cielniak

School of Computer Science
University of Lincoln
LN6 7TS Lincoln, United Kingdom
{fdayoub,tduckett,gcielniak}@lincoln.ac.uk

***Abstract*— Throughout a lifetime of operation, a mobile service robot needs to acquire, store and update its knowledge of a working environment. This includes the ability to identify and track objects in different places, as well as using this information for interaction with humans. This paper introduces a long-term updating mechanism, inspired by the modal model of human memory, to enable a mobile robot to maintain its knowledge of a changing environment. The memory model is integrated with a hybrid map that represents the global topology and local geometry of the environment, as well as the respective 3D location of objects. We aim to enable the robot to use this knowledge to help humans by suggesting the most likely locations of specific objects in its map. An experiment using omni-directional vision demonstrates the ability to track the movements of several objects in a dynamic environment over an extended period of time.**

I. Introduction

Robotic helpers and companions are a long-held dream of society. Common to all such robots is that they will share physical spaces with humans, and will thus need to deal with a dynamic and ever-changing world. This includes adapting to changes in the arrangement of objects and appearance of the environment – changes that may be spontaneous, discontinuous and unpredictable – as a result of human activities. However, almost all past research on robot mapping addresses only the initial learning of an environment, a phase which will only be a short moment in the lifetime of a service robot that may be expected to operate for many years.

The amount of sensory information to be processed in a lifetime is vast, so efficient methods are required for filtering, acquiring, storing and updating a robot's spatial-semantic knowledge of its working environment. To this end, we propose a long-term updating mechanism inspired by the classic modal model of human memory [1], as shown in Fig. 1. The memory model is integrated with a hybrid map that represents the global topology and local geometry of the environment, as well as the relative 3D location of objects. Our experimental set-up involves a robot equipped with an omni-directional vision sensor, and uses collections of local image features to represent objects as well as the background of places in the robot's map (see Fig. 2). A spherical view representation for each stored place in the map allows the robot to track the relative 3D location of objects and background image features using multi-view geometry. Background features are used as a qualitative descriptor for topological localization, while the 3D location of these features on the sphere are used for estimating the heading of the robot. After estimating the location and heading of the robot, the estimated state of objects and background features is updated using the proposed memory model.

The robot is assumed to be working for a long time in a dynamic environment where the objects change location as they are used by humans. The task that we want the robot to achieve is to track the location of a group of objects and then to give answers to questions such as: *"Where was object X the last time you have seen it?"* and *"What are the most likely locations to find object X in the map?"* To answer such questions, we conducted an experiment where the objects in a test environment were moved around inbetween visits by the robot and the respective contents of long-term memory analysed.

The rest of the paper is structured as follows. After an overview of the proposed memory model, Section II discusses related work. Section III describes our method for long-term adaptation. Section IV presents the experiments and results obtained. Finally we draw conclusions and discuss future work in Section V.

A. An Overview of the Memory Model

While a robotic memory need not be constrained by the fallibilities of human memory nor the exact details of its biological implementation, we believe that the modal model of human memory provides a natural framework for the filtering and storage of perceptual information in artificial agents such as robots. According to the basic model of Atkinson and Shiffrin [1], human memory is divided into separate sensory memory (SM), short term memory (STM) and long term memory (LTM) stores. The sensory memory contains information perceived by the senses, and selective attention determines what information moves from sensory memory to short-term memory. Through the process of rehearsal, information in STM can be committed to LTM to be retained for longer periods of time. In return, the knowledge stored in LTM affects our perception of the world, and influences what information we attend to in the

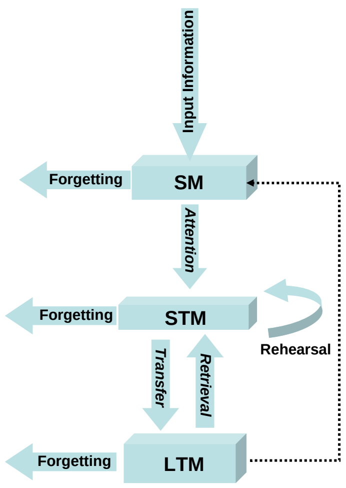

Fig. 1. An overview of the memory model.

environment. In our approach, perceptual attention includes detection of local image features and objects for subsequent processing in the memory model.

A number of the assumptions underlying this model were subsequently questioned, causing it to be further elaborated [2]. According to Tulving's model [3], LTM can be divided into declarative and procedural memory, and declarative memory is further divided into semantic and episodic memory. Episodic memory provides the capacity to remember specific events (e.g. for the purposes of this paper, specific experiences of objects and places), while semantic memory stores accumulative knowledge of the world (e.g. some generalised representation of the different episodes experienced). Forgetting plays an important role in maintaining a compact representation of the world for subsequent reasoning. Generalisation is believed to be one of the important processes involved for improving the efficiency, scalability and adaptability of cognitive systems operating in dynamic environments [4]. Our model follows this approach, maintaining long-term memories which capture the accumulative experience of places (aggregated memories of objects and background image features) rather than storing individual visits to nodes in the map.

II. Related Work

The majority of previous work on robotic mapping assume the world is static, yet nearly every actual robot environment is dynamic. Most previous approaches to mapping dynamic environments assume that the underlying structure of the environment is static, and try to separate moving objects from the stationary parts [5], [6]. Another approach tries to classify landmarks as moving or stationary, and incorporates reversible data association within a sliding window of recent observations, to allow moving objects to be included in the map estimate [7]. While these approaches mitigate some problems of static mapping algorithms, they remain unsuitable for long-term operation because they cannot handle long-term changes to the environment.

Various authors have proposed richer world models which incorporate semantic information, e.g. based on identification of objects. Nuechter et al. [8] developed an approach for labelling regions such as wall, floor, ceiling and door in 3D range scans by extracting planes and then using prior knowledge to categorize the planes. Rottmann et al. [9] developed a supervised learning approach for labelling different indoor locations such as offices, kitchens, and corridors by training a classifier based on features extracted from vision and laser range data. Other works have investigated hierarchical maps where indoor spaces such as rooms can be further decomposed according to their functions and the objects they contain [10], [11], [12].

Common to such approaches is that the semantic level of information should help to improve the robustness of robotic mapping: for example, if a living room is decomposed into a space containing chairs and sofas, it can still be recognised when the chairs and sofas have moved. However, adding semantic information to maps is not a general solution to the long-term mapping problem, since the functions of rooms, location of objects and structure of the environment may themselves change with time. This is why we are investigating adaptive memory models for long-term operation of service robots.

There is also some related work on memory models for artificial agents, for example: Peters [13] proposed an object-based memory based on the modal model for virtual agents equipped with a synthetic vision system. Vargas et al. [4] proposed hierarchical classification data mining techniques to implement forgetting and memory generalisation mechanisms in robotic companions. Mavridis and Petychakis [14] identified a list of sixteen desiderata for human-like memory systems in socially interactive robots.

III. The Method

In this section we present our method for updating the reference views of a robot's map in a changing environment while keeping track of selected objects. First, we explain how to apply the memory model to long-term robotic mapping, then we describe the map representation and the procedures for object recognition and map updating.

A. Memory Model

In our system, the memory model is used to update the visual representation of the robot's surroundings, incrementally, by gradually adding information about new stable features in the environment, while removing information about features that no longer exist. The sensory memory contains the features extracted from the current image. Then an attentional mechanism selects which information to move to STM, which is used as an intermediate store where new observations are kept for a short time. Over this time the system uses a rehearsal mechanism to select features that are more stable for transfer to LTM. In order to limit the

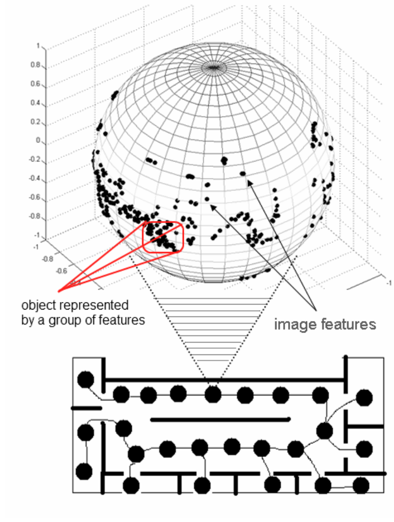

Fig. 2. Proposed Hybrid Map. The environment is represented as an adjacency graph of nodes on a topological level and each node on the metric level of the map represents the 3D location of image features on a sphere, including objects and background.

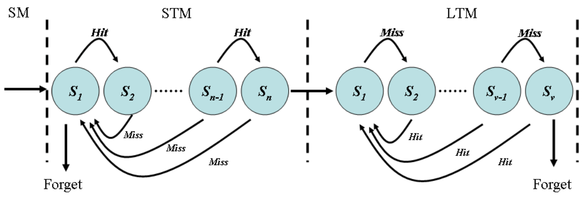

Fig. 3. The proposed multi-store memory model. SM: Sensory memory. STM: Short-term memory. LTM: Long-term memory.

overall storage requirements and adapt to changes in the environment, the system also contains a recall mechanism that forgets (i.e. removes) unused feature points in LTM. LTM is used in turn by the attentional mechanism for selecting the new sensory information to update the map.

We represent STM and LTM as finite state machines (see Fig. 3), where each memory type consists of a set of states (S_i). There is one STM and one LTM associated with each node of the map that store information about features. The LTM represents the recent stable configuration of features in the environment and these are features that are used as reference views of the map.

The rehearsal process for a stored feature in STM is the process of continually recalling information into the STM in order to memorise it. In order to transfer a feature point from STM to LTM the feature has to be seen frequently. Features enter STM from sensory memory and must progress through several intermediate states (S_1 to S_n) before transfer to LTM. Every time the robot finds the feature ("hit"), the state of the feature is moved closer to LTM. However if the feature is missing from the current view ("miss"), it is returned to the first state (S_1) or forgotten if it is already there. This policy means that spurious features should be quickly forgotten, while persistent features will be transferred to LTM.

The recall process for a stored feature in LTM first involves updating the LTM by process of feature matching. In order to remain in the LTM, a feature has to be occasionally seen. In contrast to rehearsal, features enter LTM from STM and must progress through several intermediate states (S_1 to S_v) before being forgotten. Stored features which have been seen in the current view are reset to the first state (S_1), while the state of features which have not been seen is progressed, and a feature point that passes through all states without a "hit" is forgotten. Finally, recall returns the list of new features that were not already present in the LTM (i.e. the difference in appearance between the current and reference views).

B. Map Representation

In our system, the robot's world is represented as a graph of nodes corresponding to places in the real environment. Each node in turn is represented as a set of features that describe the appearance of the environment including objects of interest. The features are provided by an omni-directional camera mounted on the robot (see Fig. 4). For local feature extraction we use the SURF algorithm [15]. In our approach, an object is represented by a collection of local features.

To represent the spatial relations between image features, and to enable efficient matching of views, we use a spherical representation for each node, where image features are projected onto a unit sphere (see Fig. 2). The position of each feature is represented by its spherical coordinates $\mathbf{x} = [\theta, \phi]^T$. This representation enables the use of features for robot self-localisation, view registration, map adaptation and object recognition.

In our experiments, we assume that an initial map of the whole environment has already been created, e.g. using an existing algorithm for topological mapping of static environments or by hand, as in our experiments. One image is selected to represent each node in the map. For each node, local features are extracted and used directly to initialise LTM, while STM for each node is initially assigned to be empty. Self-localization is carried out by comparing the current features to the reference features of each node (LTM) to estimate the current node. We apply global localization by place recognition using a simple winner-take-all strategy, although any appropriate self-localization algorithm could be applied, e.g. Markov localization.

C. Object Recognition

To enable the robot to detect important objects in the environment we use the following method based on the popular bag-of-words approach. The database of objects is first created by extracting a set of features representing the appearance of each object (again using SURF features). This way the object recognition is less sensitive to noise caused by different geometrical distortions and lighting conditions. Figure 5 shows the objects used in our experiments. The number of stored features for each object was as follows:

- Roomba box: 305.
- Cornflakes box: 259.
- Panoramic Mirror box : 147.

These features were generated from 3 views for each object.

Fig. 4. The experimental platform. An ActivMedia P3-AT robot equipped with an omnidirectional vision system.

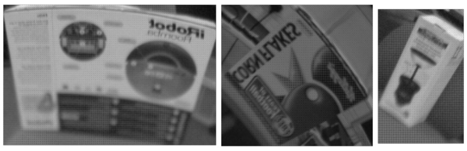

Fig. 5. The three objects used for the experiment.

Object recognition is realised by feature matching. A feature in the current view is compared to a feature in the object database by calculating the Euclidean distance between their descriptor vectors. A matching pair is detected, if its distance is smaller than a predefined threshold. When the number of feature matches exceeds another threshold, the object is detected.

Note that in our current approach we allow multiple instances of the same object class at different physical locations to be stored in memory, and we do not attempt to solve the problem of object identity resolution, i.e. resolving whether two observations taken at different times (with a significant sensory "gap" between observations) refer to the same object or different objects. Instead, we use the memory model to add new object instances (where an instance is defined as an object type plus location) and to delete object instances which have not been observed recently.

D. Map Updating

The general steps involved in our map updating mechanisms are outlined as follows:

1) *Object detection* – the "selective attention mechanism" first detects all features and then all objects that are present in the current view.
2) *Background image registration* – features corresponding to any of the detected objects are excluded, then the remaining features are used for topological localisation (determining the current map node) and registration of the current view with the corresponding reference view in LTM. Registration gives the relative orientation of the current view with respect to the stored view.
3) *Projection of observed features into map coordinates* – because new features will not be recorded in exactly the same location as the original features, all features in the current view (including both object and background features) are then projected into the spherical view representation for the current node.
4) *Memory update* – the detected objects from the current view are used to update the STM and LTM memory stores, according to the memory model described above, by either adding new object instances and deleteing object instances which have not been observed recently. At the same time, we apply a recursive filtering method to update the stored location (spherical coordinates) of these objects features.

The main details of these steps are described as follows.

After detecting all objects present in the current view, image registration is realised by finding the correspondence between background (non-object) features in the reference view and the current view. To find the correspondence between two views we are using the epipolar geometry for spherical cameras [16]. The method first estimates the so-called essential matrix $\mathbf{E}$. This matrix can be linearly solved using eight pairs (or more) of corresponding points from the two spheres [17]. In our case, the corresponding points are generated from the two views using the descriptors of the image features which will typically generate more than 8 correspondences between the two views. For this reason and the fact that the false matches will always be part of the matching process, using the RANSAC algorithm [18] is a very efficient way to minimize the effect of the outliers and find the best essential matrix to fit most of the points.

Since the robot in our case is working on a planar floor, we can simplify the process of estimating the essential matrix by restricting it to the following sparse form [19], assuming translation in x-y plane and rotation around z-axis:

$$\mathbf{E} = \begin{bmatrix} 0 & 0 & e_{13} \\ 0 & 0 & e_{23} \\ e_{31} & e_{32} & 0 \end{bmatrix}. \tag{1}$$

Based on the method introduced by Hartley and Zisserman in [20], the essential matrix is factored to give Eq. 2 which contains the rotation matrix $\mathbf{R} \in SO(3)$ and the skew-symmetric matrix $[\mathbf{T}]_\times$ of the translation vector $\mathbf{T} \in \mathbb{R}^3$:

$$\mathbf{E} = [\mathbf{T}]_\times \mathbf{R}. \tag{2}$$

This will generate multiple solutions, i.e. four possible combinations of $\mathbf{T}$ and $\mathbf{R}$. However, by applying the positive depth constraint we obtain the one solution where the reconstructed point lies outside of the two spheres [21].

After the estimation of $\mathbf{T}$ and $\mathbf{R}$, we can find the relative direction α and rotation γ between two views where:

$$\alpha = atan(\mathbf{T}[y], \mathbf{T}[x]), \tag{3}$$

and

$$\mathbf{R} = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

In order to adapt the map, the feature points which need to be moved to the STM and LTM stores of each node should be transferred on the reference sphere as if these features were seen from the same point where the node was first created. To achieve this, we reconstruct the 3D position of feature points shared between the current and previous views. The 3D position of the shared points can be determined to unknown scale as the norm of the translation vector is fixed to unity. These points are divided into three groups: the points which already exist in the LTM store of the node, the points which already exist in the STM store of the node and the new points which need to be added to the STM.

In order to add the new features to STM in their correct position on the sphere we use a simplified version of what is known as multibaseline stereo [22]. In our case, we only use two stereo pairs between three views: the reference view, the current view and the view from the previous visit to the node. The views are captured in different visits to the node and we are not interested in recovering a 3-D map for a large scene; instead we want to update a single spherical view by adding new feature points to it. Linear triangulation is used to obtain the desired 3D position of a point. More details of the linear triangulation approach can be found in [20].

Features from the current view are then used to update the robot's memory, as described in Section III-A. The estimate state $\mathbf{x} = [\theta, \phi]^T$ of each feature is updated using an Unscented Kalman Filter (UKF) [23] in which a small number of carefully chosen sample points is propagated in each estimation step. The observation model which relates 3D measurements to the state vector involves the nonlinear mapping:

$$\mathbf{z} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \sin(\theta)\sin(\phi) \\ \cos(\theta)\sin(\phi) \\ \cos(\phi) \end{bmatrix}, \quad (5)$$

where z is the observation vector. For a state space with dimension L, $2L+1$ points are selected such that their sample mean is the state vector and their covariance is the process covariance (so in our experiments, 5 points were used). The nonlinear function is applied to each point in turn to yield a cloud of transformed points, which provide a compact parameterization of the underlying distribution.

The memory update for objects is slightly more involved. The robot goes through all the instances for each detected object in LTM and finds the matched points. If the instance is still in the same location, the matched points will overlap or be close to each other (due to noise in the estimation of $\mathbf{R}$ and $\mathbf{T}$). But if the matched points are far apart from all instances in memory, a new object instance will be created and added to STM. To determine whether two object instances occupy the same location, we convert the points to spherical coordinates and find the root mean square deviation between the horizontal angles θ of the matched points, using a match threshold of 2.5 degrees in our experiments. Again applying the memory model in Section III-A, all matched object instances have their stage in LTM reset to state 1, while all unmatched instances advance towards being forgotten.

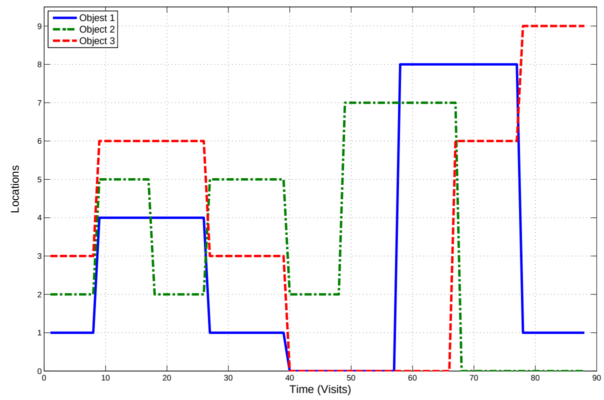

Fig. 6. Ground truth information for the experiment, showing the nine discrete locations of the dynamic objects over time.

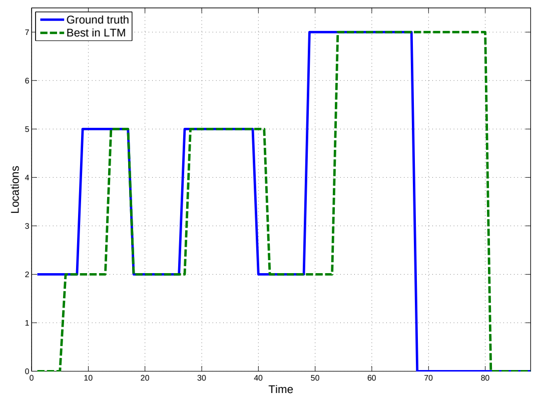

Fig. 7. Most likely location in LTM of object 2 against ground truth. An instance of the object first appears at location L2, which enters the STM of the node and appears in LTM after 5 visits. Then an instance of the object appears in a new location L5, and the robot starts to forget the instance at L2, with the near instance appearing in LTM after 6 visits. Then the object reappears at L2 at visit 17 – this time the instance does not need to go through STM again as it was not yet forgotten from LTM.

IV. Experiment

To investigate the proposed system, we conducted a preliminary experiment in our robotics lab using an ActivMedia P3-AT robot equipped with a GigE progressive camera (Jai TMC-4100GE, 4.2 megapixels) with a curved mirror from 0-360.com (see Fig. 4). For previous experiments involving long-term updating of the topological and metric levels of the map representation please see [24] – here we test the ability of the system to track objects over time.

In this experiment we use one map node only, and we assume that the robot has already determined its current node by topological localization, and then starts to observe the objects in that node. We chose three objects (see Fig. 5) and extracted a group of SURF features from a set of images for each object. The test data consisted of 88 images

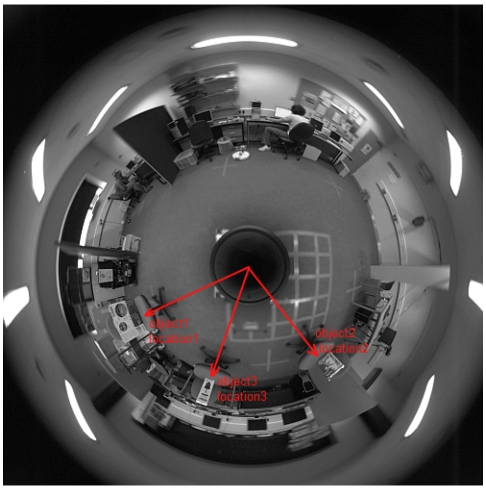

Fig. 8. Image 5 in the recorded sequence.

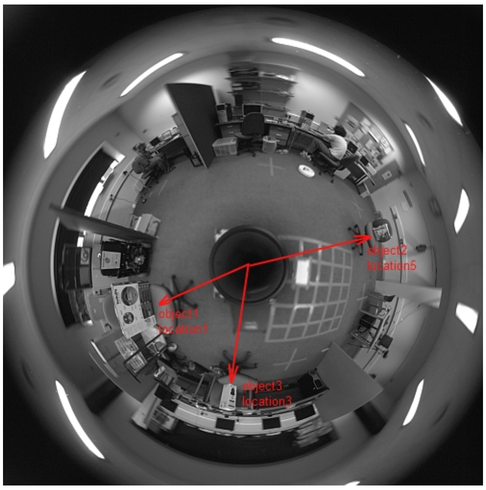

Fig. 9. Image 35 in the recorded sequence.

recorded from random locations of the robot in our lab, to simulate different "visits" to the same place, i.e. so that there would be random sensory "gaps" between observations (meaning that traditional methods for data association in tracking would fail). At 8 randomly chosen occasions during the experiment, we manually changed the locations of some objects, sometimes temporarily removing the object from the room. There were three different location for each object, giving a total of 9 different locations. Fig. 6 shows the location of each object over the 88 "visits" of the robot to the node. See also Figs. 8 and 9 for two recorded images, showing also the location of the dynamic objects.

The memory model was tested with 5 stages for STM and 15 for LTM. Sometimes the robot was not able to detect the objects due to occlusion or a low number of matched features. To answer the question *"What is the most likely location of object X?"*, we interpret the robot's memory by looking for the corresponding object instance in LTM that is closest to stage 1 (i.e. the most recently observed instance in LTM). Fig. 7 shows a trace of the most likely location for object 2 compared to the ground truth data.

For example, in the case of object 2 at visit number 22, there were two instances of the object in LTM at locations $L1$ and $L2$, each at different stages in the memory. Instance $L1$ was at stage 5 due to the forgetting effect, while instance $L2$ was at stage 1, indicating that the object was seen here in the last visit. As another example, if we look at the state of the memory for object number 3 at visit number 70, we find that there is an old instance of this object in LTM indicating that the object has not been seen for more than 14 visits to the node, but STM contains an instance of the object at stage 3, since the object had reappeared in the node recently.

V. Conclusion

This paper presented an object-based semantic memory for mobile service robots, augmenting our previous results in long-term operation for the topological and metric levels of the robot's map [24]. The results show that the semantic memory follows the ground truth location of objects in the test environment with a small time lag, with some variation due to noise in perception. Future work would include further integration of the different memory and reasoning systems required for long-term operation of a service robot, for example, including inference of the functions of rooms and behaviours of persons (monitoring interactions of people, places and objects) as they change over time.

References

[1] R. Atkinson and R. Shiffrin, "Human memory: A proposed system and its control processes," *In K.W. Spence & J.T. Spence (Eds.), The Psychology of Learning and Motivation*, vol. 2, pp. 89–195, 1968.

[2] A. Baddeley, M. W. Eysenck, and M. C. Anderson, *Memory*. Psychology Press, 2009.

[3] E. Tulving, "Episodic and semantic memory," in *Organization of Memory*. New York: Academic Press, 1972, p. 89101.

[4] P. A. Vargas, A. A. Freitas, M. Lim, S. Enz, W. Ching Ho, and R. Aylett, "Forgetting and generalisation in memory modelling for robot companions: a data mining approach," in *Proc. Human Memory for Artificial Agents Symposium (AISB 2010)*, De Montfort University, Leicester, UK, 2010.

[5] D. Anguelov, R. Biswas, D. Koller, B. Limketkai, S. Sanner, and S. Thrun, "Learning hierachical objects maps of non-stationary environments with mobile robots," in *Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI)*, 2002.

[6] C.-W. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking if moving objects: Theory and results from a ground vehicle in crowded urban areas," in *ICRA*, 2003.

[7] C. Bibby and I. Reid, "Simultaneous localisation and mapping in dynamic environments (SLAMIDE) with reversible data association," in *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.

[8] A. Nuechter, H. Surmann, K. Lingemann, and J. Hertzberg, "Semantic analysis of scanned 3d indoor environments," in *Proceedings of the 8th International Fall Workshop on Vision, Modelling, and Visualization 2003 (VMV '03)*. Munich, Germany: IOS Press, Nov. 2003, pp. 215–222.

[9] A. Rottmann, O. Martínez Mozos, C. Stachniss, and W. Burgard, "Place classification of indoor environments with mobile robots using boosting," in *Proc. National Conference on Artificial Intelligence (AAAI)*, Pittsburgh, PA, USA, 2005.

[10] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. Fernández-Madrigal, and J. González, "Multi-hierarchical semantic maps for mobile robotics," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Edmonton, CA, 2005, pp. 3492–3497.

[11] S. Vasudevan and R. Siegwart, "Bayesian space conceptualization and place classification for semantic maps in mobile robotics," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 522–537, June 2008.

[12] H. Zender, O. M. Mozos, P. Jensfelt, G.-J. M. Kruijff, and W. Burgard, "Conceptual spatial representations for indoor mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 493–502, June 2008.

[13] C. Peters, "An object-based memory for supporting attentive virtual agents," in *Proc. Human Memory for Artificial Agents Symposium (AISB 2010)*, De Montfort University, Leicester, UK, 2010.

[14] N. Mavridis and M. Petychakis, "Human-like memory systems for interactive robots: Desiderata and two case studies utilizing grounded situation models and online social networking," in *Proc. Human Memory for Artificial Agents Symposium (AISB 2010)*, De Montfort University, Leicester, UK, 2010.

[15] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," 2006.

[16] T. Akihiko and I. Atsushi, "Multiple view geometry for spherical cameras," *IEIC Technical Report (Institute of Electronics, Information and Communication Engineers)*, vol. 105, pp. 29–34, 2005.

[17] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, pp. 133–135, 1981.

[18] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, pp. 381–395, 1981.

[19] J. Košecká, F. Li, and X. Yang, "Global localization and relative positioning based on scale-invariant keypoints," *Robotics and Autonomous Systems*, vol. 52, pp. 27–38, 2005.

[20] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, Mar. 2004.

[21] B. K. Horn, "Relative orientation," *International Journal of Computer Vision*, vol. 4, no. 1, p. 5978, 1990.

[22] S. B. Kang and R. Szeliski, "3-D scene data recovery using omnidirectional multibaseline stereo," *International Journal of Computer Vision*, vol. 25, pp. 167–183, 1997.

[23] S. J. Julier and J. K. Uhlmann, "A new extension of the kalman filter to nonlinear systems," in *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, vol. 3, 1997, p. 26.

[24] F. Dayoub, G. Cielniak, and T. Duckett, "Long-term experiments with an adaptive spherical view representation for navigation in changing environments," *Robotics and Autonomous Systems (to appear)*, 2010.