Proceedings of the Remembering Who We Are – Human Memory for Artificial Agents Symposium, Mei Yii Lim and Wan Ching Ho (Eds.), at the AISB 2010 convention, 29 March – 1 April 2010, De Montfort University, Leicester, UK

Short- and Long-term Adaptation of Visual Place Memories for Mobile Robots

Feras Dayoub and **Tom Duckett** and **Grzegorz Cielniak**[1]

Abstract. This paper presents a robotic implementation of a human-inspired memory model for long-term adaptation of spatial maps for navigation in changing environments. The robot uses an appearance-based representation of its workplace as a map, where the current view and the map are used to estimate the robots current position in the environment. Due to the nature of real-world environments such as houses and offices, where the appearance keeps changing, the map may become out of date after some time. To solve this problem the robot needs to adapt the map continually in response to the changing appearance of the environment. In this work we use local features extracted from panoramic images to represent the appearance of the environment. Adopting concepts of short-term and long-term memory, our method updates the group of feature points for the image representation of a particular place. Experiments using robot sensor data collected over a period of 2 months show that the implemented model is able to adapt successfully to changes.

1 INTRODUCTION

Robotic helpers and companions have become a major topic for current research. This trend has been facilitated by the increased performance of modern processors and the emergence of low-cost, high performance sensors such as digital cameras, together with recent advances in academic disciplines including computer vision, artificial intelligence and mobile robotics. Simple robots such as vacuum cleaners and lawn mowers are already available to consumers, while many predict that personal service robots will become as ubiquitous in the home and office of the future as the PCs and mobile phones of today. 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.

A mobile service robot's first tour of a new environment will probably be guided by a human, for example, to explain the layout and function of different places, just as a human employee would be introduced to a new workplace. One problem for the robot is to build a sufficiently detailed map to begin working in the new environment. However, the initial learning of an environment will only be a short moment in the lifetime of a service robot that is expected to operate for many years.

Most of the existing work on localization and mapping assumes that the environment where the robot works is static. However, this assumption does not hold for many real environments. For example, the appearance of a room in a house is not static over time: new objects are sometimes added, existing objects like pictures or carpets may be changed or moved, and old objects may be removed. Some of these changes occur very often, such as moving chairs and cushions, etc. These transient variations need to be excluded from the long-term representation of the appearance of the environment.

In this paper, we propose a method to update the reference views of a topological map for a changing environment, which is represented as an adjacency graph where the nodes of the graph represent distinctive places and the edges represent the transitions between places. Adopting concepts of short-term and long-term memory inspired by an information processing model (see Fig. 1) based on the multi-store model of human memory proposed by Atkinson and Shiffrin [2], our method updates a group of feature points for the reference view of a particular place. Thus the reference views are adapted to represent the information about the new appearance of the location.

The reminder of this paper is structured as follows. After dis-

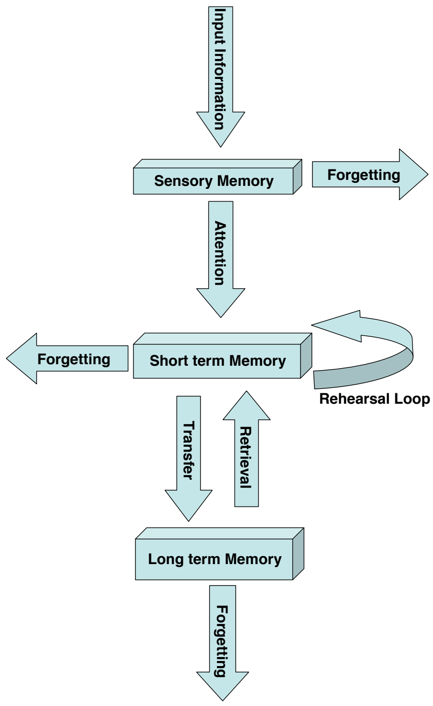

Figure 1. The Information Processing Model.

[1] University of Lincoln, United Kingdom, email: fdayoub,tduckett,gcielniak@lincoln.ac.uk

cussing related work in Section 3, we introduce our memory model and the adaptation mechanism in Section 4. Section 5 presents the experiments and results obtained. Finally we draw conclusions and discuss future work.

2 ADAPTABILITY AND THE HUMAN MEMORY

From a biological point of view, an organism is adaptable if it can survive significant changes in its environment, spread to new habitats, and come up with novel solutions to its surroundings[2]. For humans, adaptability is a key factor for survival and depends heavily on accurate evaluation of environmental changes. This accuracy is a product of our personal experience which is in turn collected and stored in our memory.

When talking about mobile robots which can work among us and inside our everyday environments, the biologically inspired approach sounds very appealing. This is because these robots will work in the same context in which the biological systems themselves operate and will face the same challenges. One of the important problems which need to be solved before we can see these robots sharing our rooms and offices is the ability to adapt to changes in the working space, from the appearance of a room to the behaviour of people.

Nature has had millions of years to evolve and has lead to very successful systems making life able to survive under very hard conditions. Based on this observation one can suggest that we should copy nature's design directly and implement these systems. But anyone with a biological background will know that each biological system has a very long inheritance of tinkering which has produced very complex structures and implementing this is not the right way to go. Instead we should be inspired by biology by discovering the general principles on which these systems are built and then applying such themes in the heart of the solution.

Human memory is not yet fully understood and many of the ideas about how it works are still controversial. Among the ideas which is widely accepted is to look at the memory in terms of multiple stores of information and a set of processes that act on these stores. In this work, we adopt and bring this multi-store theory to the domain of mobile robot mapping and localization, where we use it to enable the robot to adapt its internal representation of the world over time in response to the changing of the appearance of its surroundings.

3 RELATED WORK

Most previous approaches to mapping in dynamic environments assume that the underlying structure of the environment is static, and try to separate moving objects from the stationary parts. The dynamic effects are often treated merely as measurement outliers. Alternatively moving objects are detected and tracked separately using multi-target tracking techniques [15]. Another recent 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 into the SLAM estimate [4]. In general, while these approaches mitigate some problems of classical SLAM algorithms, they cannot handle long-term changes to the structure of an environment.

Several authors have investigated mapping systems that incorporate simple forgetting mechanisms based on recency weighting. For example, Yamauchi [16] developed a system that learns and updates the topology of a map during runtime based on successful and unsuccessful attempts to traverse a given link. Andrade-Cetto [1] developed an EKF-based mapping system that is able to forget landmarks that have disappeared, where an existence state associated with each landmark measures how often it has been seen. However, none of these methods are general enough to handle environmental changes occurring at different rates, nor has the long-term robustness of these approach been demonstrated in real world environments.

This paper tackles the problem of working in a changing environment by using an appearance based approach to represent the workplace of the robot. An early approach for appearance-based mapping and localization was published in [13] where the operational area of the robot is represented as a graph. The nodes in this graph represent distinctive places and the edges represent the transitions between places. This approach consists of two stages. In the first stage the robot is driven through its operational area to learn a model of the environment, by taking a sequence of images in certain places and then creating a topological map from these images. In the second stage the robot uses the map to find its current position, i.e. the node which is most similar to the current view.

Two different approaches to measure the similarity between images have been presented in the literature: global and local methods. The global methods capture global properties of the image using approaches based on colour histograms [7], principal component analysis (PCA) [14], Fourier transform [10], etc. The local methods extract local properties from the image and produce a group of landmark features. Local feature including SIFT [8], SURF [3], MSER [9], etc., are used to find the similarity between images. The local methods have been shown to be more reliable and robust to illumination and viewpoint changes, thanks to the feature descriptors that are built using a local region around selected feature points. Each feature is described by a high-dimensional vector, which has high invariance to image translation, scaling and rotation, and partial invariance to illumination changes and affine projection.

Using local image feature descriptors, the similarity between two images can be measured by finding the correspondences between the features in the two images. This can be done by finding the closest feature in the feature descriptor space. However, the method can be time consuming if the number of images in the map is large and the search is done linearly. Using a text retrieval approach, Sivic and Zisserman [12] presented a very fast retrieval system using a visual vocabulary. Nister and Stewenius [11] extended the ability of the system by using hierarchical K-means clustering to improve the performance, so that Fraundorfer et al. [5] were able to implement global localisation in real-time.

4 LONG-TERM ADAPTATION USING MEMORY STORES

For any mobile robot using a topological map to represent its working space, the map could become out-of-date after some time in a changing environment. A naive solution to this problem would be simply to replace the image representation for each node in the topological map from time to time, in order to reflect the changed appearance of the corresponding location. Provided that the robot is correctly localized, this approach would enable the robot to remove out-of-date information from the map. However, it could also remove useful features due to temporary occlusions, and could lead to catastrophic results in the case of localization errors. A better solution would be to update the image representation of a node incrementally, by gradually adding information about new stable features in the en-

[2] http://humanorigins.si.edu/aop/adaptability.htm

vironment, while removing information about features that no longer exist. In our approach, each node is represented by a group of features (using SURF in these experiments, though other features could be used). This group of features is updated over time by adding persistent new features and removing older ones that are no longer used. In order to achieve this we adopted short-term and long-term memory concepts based on the multi-store model of human memory proposed by Atkinson and Shiffrin [2]. This model, which forms the basis of modern memory theories, divides human memory into three stores:

- sensory memory (SM),
- short-term memory (STM),
- long-term memory (LTM).

4.1 Recall, Rehearsal, Transfer

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 environment.

Applying these concepts to our approach for robot mapping, the sensory memory will contain 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 overall storage requirements and adapt to changes in the environment, the system also contains a recall mechanism that forgets unused feature points in LTM by removing these features from the node. LTM is used in turn by the attentional mechanism for selecting the new sensory information to update the map.

We model the world as a set of discrete places. In our experiments, omni-directional vision is used to provide the features for localization and mapping. We assume that an initial map of the whole environment has already been created by the robot, e.g. using an existing algorithm for topological mapping of static environments. (In this work we selected the places by hand.)

Each place has two memory stores: STM, which is a temporary stage, and LTM, which provides the reference views in the map used for self-localization. We assume that the robot is able to self-localize by matching features extracted from the current view to the stored reference views. The purpose of our algorithm presented here is to maintain up-to-date reference views for the nodes in the map, using recall and rehearsal concepts inspired by human memory. To initialise the map, the image data from the robot's first tour of the environment is used. 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.

Algorithm 1 Update the reference view.

```
Definitions:
CrrNode: The reference view of the current node.
CrrView: The current view for the current node.
CrrSTM: The current STM for the current node.
STMlng: The maximum number of states in the STM.
LTMlng: The maximum number of states in the LTM.
newFP: The difference between the CrrView and CrrNode.
——————————————————
for (every visit to the node ) {
        newFP = recall( CrrNode , CurrView , LTMlng )
        rehearse( CrrSTM, newFP , STMlng )
  }
```

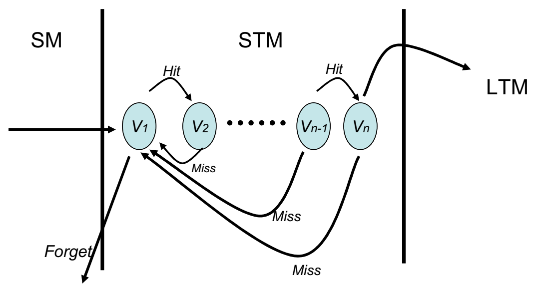

Figure 2. The rehearsal stage in the STM.

Algorithm 2 The rehearsal stage in the STM.

```
for (every feature in CrrSTM ){
    if (feature in newFP){
       Move the feature to the next state.
       if (feature state > STMlng){
          Move the feature to the CrrNode.
          Remove the feature from CrrSTM.
       }
       else if (feature in the first state){
              Remove the feature from CrrSTM.
       else
              Reset the feature to the first state.
       }
    }
}
for (every feature in newFP ){
    if (feature was not in CrrSTM){
       Add the feature to CrrSTM in the first state.
    }
}
```

Thereafter, every time the robot visits an existing node, the following steps are carried out. Feature points are extracted from the current view. Self-localization is carried out by comparing the current features to the reference features of each node (LTM) to estimate the current node. In our case, 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. After localization, the current features are used in the recall stage for updating the LTM of the current node. Only new features which do not match any feature in LTM are used in the rehearsal stage. Algorithm 1 describes the two main stages; (1) recall, where the difference in appearance between the reference and current views is computed, and (2) rehearsal, where this difference is used to update STM and commit persistent new features of the location to LTM. Algorithm 2 shows the rehearsal process for a stored feature in STM, which is also represented as a finite state machine in Fig. 2. This stage represents what Atkinson and Schiffrin called rehearsal in their memory model (Fig. 1), i.e. the process of continually recalling information into the STM in order to memorise it. In order to

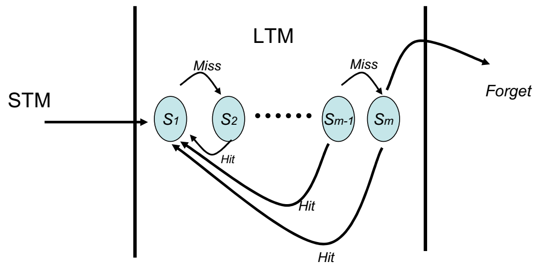

Figure 3. The recall stage in the LTM.

Algorithm 3 The recall stage in the LTM.

```
newFP = []
for (every feature in CrrNode){
    if (feature in CrrView){
       Rest the feature to the first state.
    else
       Move the feature to the next state.
    }
    if (feature state > LTMlng){
       Remove the feature from CrrNode.
    }
}
for (every feature in CrrView ){
    if (feature not in CrrNode){
       Add the feature to newFP.
    }
}
return newFP
```

transfer a feature point from STM to LTM the feature has to be seen frequently in that node. Features enter STM from sensory memory and must progress through several intermediate states (V_1 to V_n) before transfer to LTM. Every time the robot visits the node and 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 (V_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.

Algorithm 3 shows the recall process for a stored feature in LTM, which is also represented as finite state machine in Fig. 3. This process first involves updating the LTM by matching the reference view to the current view. In order to remain in the LTM, a feature has to be seen occasionally in that node. In contrast to rehearsal, features enter LTM from STM and must progress through several intermediate states (S_1 to S_m) 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).

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

5 EXPERIMENTS AND RESULTS

To investigate our method of updating the reference views in a topological map, we conducted two experiments. In the first experiment we tested the system for a single node represented by a view of an office room. In the second experiment we used an image data set recorded over a period of approximately two months from a real changing environment.

Our experimental platform was 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 local feature extraction we use the SURF algorithm. This algorithm extracts local features from the scale-space of the image based on the Hessian matrix and approximates the second order Gaussian derivatives with box filters. A fast non-maximum suppression algorithm is also used. The resulting algorithm has a good performance in the extraction process and a high accuracy. For more details, see [3].

5.1 Testing the system using data from a manually changing environment

For the first experiment, the robot first takes a reference view in the middle of a room. Then the robot is moved back and forth covering approximately 1 m on each side of the place where the reference view was recorded, while the appearance of the room is changed manually. The changes involve the arrangement of the objects inside the room, including adding new objects like boxes and posters, removing existing objects individually and also covering them with movable office dividers for certain periods. Fig. 6 shows two sample views from the room where the experiment was conducted.

While the robot was moving, a set of 251 images was recorded to capture the changing appearance of the room. After capturing each image, SURF image features were extracted and saved to a file. Using these files, the memory model was then tested offline.

The average number of features extracted from each image was 600.0 ± 56.1. The mean number of matched points between the images of the dataset and the static reference view was 45.2 ± 25.4 points whereas it was 98.8 ± 18.7 when the dynamic view was used, indicating that the adaptive reference view represents the current state of the environment more faithfully than the static view.

Fig. 5 shows how the number of feature points changes inside the LTM and STM during the experiment, using 8 stages for the LTM and 3 stages for the STM. The size of the STM changes rapidly where the features enter and stay for a short time whereas the change on the LTM size is slow as the features stay longer.

Proceedings of the Remembering Who We Are – Human Memory for Artificial Agents Symposium, Mei Yii Lim and Wan Ching Ho (Eds.), at the AISB 2010 convention, 29 March – 1 April 2010, De Montfort University, Leicester, UK

Figure 6. Views of the room during the experiment. Various objects were added, removed or changed manually to test the adaptive capabilities of the system.

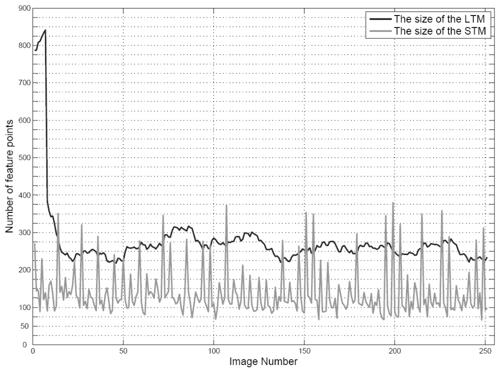

Figure 5. Number of feature points into the LTM (with 8 stages) and the STM (with 3 stages) during the experiment.

5.2 Long-term experiment using data from a real changing environment

This experiment was carried out in a real changing environment of an office floor at the University of Lincoln. This area is used for student and staff activities.

Over a period of approximately two months, we took the robot on tours between the offices while recording a set of images. The result was 36 tours (one tour per day) generating 6161 images with approximately 35cm between consecutive images. We used the first tour to created a topological map by choosing 21 omni-directional images from different places to form the reference views for the nodes. Fig. 7 shows the ground truth positions of the recorded images. Each colour represents the group of images which were assigned to the same node. The crosses represent the reference view for each node. In order to obtain the ground truth data we used the GMapping library [6]. The GMapping algorithm provides a Simultaneous Localization and Mapping (SLAM) solution based on a Rao-Blackwellized particle filter. The output of the algorithm is an estimate of the robot trajectory along with an occupancy grid map of the environment. The first tour is used for creating the topological map. For each subsequent tour, we estimated the robot's true trajectory through the environment then we aligned the output map with the first tour map in order to transform the ground truth navigation trajectories into a common reference frame. We emphasize that our method does not build or require a metric map of the environment. We only use this information for ground-truth validation.

We tested our method for adapting the reference views in the map using the collected images. We used a global localization method based on place recognition (winner-takes-all). We carried out the tests using the memory model with 8 stages in the LTM and 2 stages in the STM.

The results show that the failure rate of the global localization was around 1.5% when the adaptive approach was used whereas the failure rate for the static one was 3.5%.

Fig. 8 shows the cumulative percentage of matched points used for the global localization. The red line illustrates the results when the static reference views were used for the map whereas the green line corresponds to the adaptive views. The adaptive approach provides better matching scores leading to better localization performance and also better navigation performance. For example, 43% of the time the number of matched points between the reference view and the current image from the robot was below 35 points for the static views, whereas this happened only 24% of the time with the adaptive vews.

6 CONCLUSIONS AND FUTURE WORK

This paper introduced an adaptation mechanism for topological localization methods based on features extracted from images that represent the appearance of the environment. Our method updates the reference views of the nodes in the map and keeps track of the changes in the appearance of the environment, while the robot is

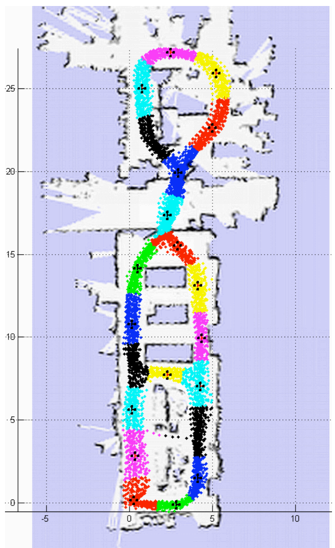

Figure 7. Ground truth positions of the recorded images obtained from the laser-corrected odometry. The constructed map consists of 21 nodes, each colour represents the group of images which were assigned to the same node. The crosses represent reference views for each node.

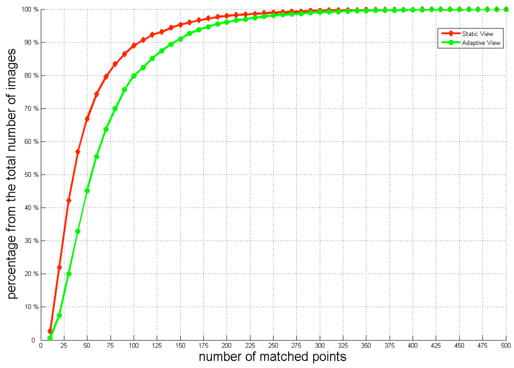

Figure 8. The cumulative percentage of matched points used for the global localization. The red line illustrates the results when the static reference views were used for the map whereas the green line corresponds to the adaptive views.

working over long periods of time. To achieve this we adopted short-term and long-term memory concepts to adapt the reference views of the nodes in response to the dynamics of the environment. To test our method, we conducted two experiments. The first one was in an office room where we manually changed the appearance of the place. For the second experiment we used data recorded from a real dynamic environment over two months. In both experiments the method gave improved results over static mapping.

In this work, the number of the stages in LTM and STM were determined empirically based on the recorded sensor data. As a future work, the number of the stages could be learned depending on the dynamics of the real environment. The attention mechanism could be improved by adding real-time tracking of features in the scene to filter out spurious features due to noise or temporary occlusion. The adaptive capability of the map could be further extended to the topological level, by making the robot able to add or remove nodes and links from the map.

7 DISCUSSION

Memory is a process by which we can store and retrieve pieces of information. In this work, the pieces of information are the image features. We used these features to recognize a place in a topological map by finding the correspondences between the features in the LTM and the current view, and also adapting the group of features in the LTM in response to the changing appearance of the node in the map through the recall and rehearsal processes. These image features have very interesting properties. The descriptor of each feature is a signature making the matching process to find these features invariant to scale, orientation, affine distortion and partially invariant to illumination changes. These properties can open the door to use these features in a higher level of representation for the environment. The topological map, as it has been presented, is a coarse level of representation for the environment which limits the possibilities to interact with the robot. For example, we can ask the robot to go from A to B but we cannot ask the robot to search for a specific object in the map. For these kind of tasks we need what is called a semantic map, where the map will contain, in addition to the spatial information, information about the objects which share the space. In our case, the semantic level can be built using the same image features. If we extract image features from an omnidirectional image taken from a room inside a home environment, we can notice that the features are not distributed uniformly over the space and certain areas have more density than others due to how the objects are arranged inside the room. Each area could contain one real object or multiple parts of objects which generate the features. In order to describe a place in the environment, the robot can use these groups as what we will call abstract objects (AO). These AO's can be tracked over time using the memory model. Each time an abstract object reaches the LTM memory, the robot can ask a human to associate it with a semantic label if the features of the AO came from a real object or ignore the AO if the features do not correspond to a single object. In this way the robot will start to build a map of the objects over time and also, thanks to the memory model, be able to forget objects which have been disappeared from view.

REFERENCES

[1] J. Andrade-Cetto and A. Sanfeliu, 'Concurrent map building and localization in indoor dynamic environments', *Int. J. Pattern Recognition and Artificial Intelligence*, **16**(3), 361–374, (2002).
[2] R.C. Atkinson and R.M. Shiffrin, 'Human memory: A proposed system and its control processes', *In K.W. Spence & J.T. Spence (Eds.), The Psychology of Learning and Motivation*, **2**, 89–195, (1968).
[3] H. Bay, T. Tuytelaars, and L. Van Gool, 'SURF: Speeded up robust features', in *Proc. European Conference on Computer Vision (ECCV)*, (2006).
[4] C. Bibby and I. Reid, 'Simultaneous localisation and mapping in dynamic environments (SLAMIDE) with reversible data association', in *Proc. Robotics: Science and Systems (RSS)*, Atlanta, GA, USA, (June 2007).
[5] C. Fraundorfer, F. Engels and D. Nister, 'Topological mapping, localization and navigation using image collections', in *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS)*, (2007).
[6] G. Grisetti, C. Stachniss, and W. Burgard, 'Improved techniques for grid mapping with Rao-Blackwellized particle filters', *IEEE Transactions on Robotics*, **23**(1), 34–46, (2007).
[7] H.M. Gross, A. Koenig, C. Schroeter, and H.J. Boehme, 'Omnivision-based probabilistic self-localization for a mobile shopping assistant continued', in *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS)*, (2003).
[8] D.G. Lowe, 'Object recognition from local scale-invariant features', in *Proc. IEEE International Conference on Computer Vision (ICCV)*, (1999).
[9] J. Matas, O. Chum, M. Urban, and T. Pajdla, 'Robust wide-baseline stereo from maximally stable extremal regions', *Image and Vision Computing*, **22**(10), 761–767, (2004).
[10] E. Menegatti, M. Zoccarato, E. Pagello, and H. Ishiguro, 'Image-based Monte Carlo localisation with omnidirectional images', *Robotics and Autonomous Systems*, **48**(1), 17–30, (2004).
[11] D. Nister and H. Stewenius, 'Scalable recognition with a vocabulary tree', in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2006).
[12] J. Sivic and A. Zisserman, 'Video Google: a text retrieval approach to object matching in videos', in *Proc. IEEE International Conference on Computer Vision (ICCV)*, (2003).
[13] I. Ulrich and I. Nourbakhsh, 'Appearance-based place recognition for topological localization', in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, (2000).
[14] N. Vlassis, B. Terwijn, and B. Krose, 'Auxiliary particle filter robot localization from high-dimensional sensor observations', in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, (2002).
[15] C.-W. Wang, C. Thorpe, and S. Thrun, 'Online SLAM with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas', in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 842–849, Taipei, Taiwan, (2003).
[16] Brian Yamauchi and Randall Beer, 'Spatial learning for navigation in dynamic environments', *IEEE Trans. Systems, Man and Cybernetics*, **26**(3), 496–505, (1996).