Eight weeks of episodic visual navigation inside a non-stationary environment using adaptive spherical views*

Feras Dayoub, Grzegorz Cielniak and Tom Duckett

Abstract This paper presents a long-term experiment where a mobile robot uses adaptive spherical views to localize itself and navigate inside a non-stationary office environment. The office contains seven members of staff and experiences a continuous change in its appearance over time due to their daily activities. The experiment runs as an episodic navigation task in the office over a period of eight weeks. The spherical views are stored in the nodes of a pose graph and they are updated in response to the changes in the environment. The updating mechanism is inspired by the concepts of long- and short-term memories. The experimental evaluation is done using three performance metrics which evaluate the quality of both the adaptive spherical views and the navigation over time.

1 Introduction

Functional and useful mobile service robots require the ability to share physical spaces with humans, and need to deal with a dynamic and ever-changing world. These changes are mainly caused by human activities making them spontaneous, discontinuous and unpredictable. This includes changes in the structure of the environment as well as its appearance (e.g. rearrangement of the furniture or changing the colour of a curtain).

In order to maintain an up-to-date inner representation of the world, robots can use their continuous stream of sensory information, which reflects the momentary status of their surroundings. However, the amount of sensory information to be pro-

* An extended abstract of this paper appeared as [6]

Feras Dayoub with CyPhy lab, Queensland University of Technology, Brisbane, Australia, e-mail: feras.dayoub@qut.edu.au
Grzegorz Cielniak and Tom Duckett with University of Lincoln, Lincoln, UK

cessed in a lifetime is vast; therefore, efficient methods are required for filtering, storing and updating this information over time.

One possible solution for handling this large amount of sensory information can be inspired by the concepts of short- and long-term stores of the human memory. 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 useful framework for the filtering and storage of perceptual information in artificial agents such as robots.

This paper uses a long-term map-updating mechanism inspired by the multi-store model of human memory [7] for the application of visual navigation. The map consists of an adjacency graph of nodes on a global level, and each node on the local level of the map represents a spherical view of image features extracted from an omnidirectional image of the node. The spherical views provide both an appearance signature for the nodes, which the robot uses to localise itself in the environment, and heading information when the robot uses the map for visual navigation. The paper presents an evaluation of the navigation performance within a typical office environment over a period of eight weeks. These metrics are used to evaluate the effect of the learning and forgetting processes on the quality of the map over time.

The rest of the paper is structured as follows. Section 2 discusses related work. Section 3 presents an overview of the proposed memory model, Section 4 describes our method for long-term adaptation and visual navigation. Section 5 discusses the performance evaluation metrics. Section 6 presents the experiments and results obtained. Finally we draw conclusions in Section 7.

2 Related Work

Although nearly every actual robot real-life environment is dynamic, the majority of previous work on robotic mapping assumes that the world is static. Whereas, most previous approaches that consider mapping dynamic environments assume that the underlying structure of the environment is static, and then try to separate moving objects from stationary parts by treating the dynamic effects as measurement outliers [8, 12, 10]. Alternatively, many authors try to improve the robustness of the mapping process by detecting and tracking moving objects separately [18, 15, 13].

Considering that the environment consists of static and dynamic objects, other approaches build two maps, one for the static parts and one for dynamic elements. The complete state of the environment is obtained by merging the two maps [19]. Other approaches try to maintain one map for both dynamic and static landmarks, by classifying landmarks as dynamic or stationary using a probabilistic framework. Movements of the dynamic landmarks are observed and included in the estimation process of the map [4].

Several authors have investigated mapping systems that incorporate simple forgetting mechanisms based on recency weighting. Andrade-Cetto and Sanfeliu [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 approaches been demonstrated in real world environments. Using a map learning and forgetting mechanism, Biber and Duckett [5] introduced an approach to update a laser based map which represents the environment at different timescales, with older memories fading at different rates. They used samples and robust statistics to handle noise and contradicting measurements produced by environment changes.

Very recently [17] proposed a method to update a laser-based metric map. The method uses a pose graph SLAM approach to optimise the trajectory of the robot and produce the map. In order to update the map, the robot compares its current laser scan view with scans stored from previous passes through the same sections of the environment. The author makes the assumption that the environment contains only low dynamic objects, i.e. objects that move only outside of the robot's view, which makes the environment static when the robot is passing through it. However, in a dynamic environment, changes can occur while the robot is operating inside the environment. As such, these changes need to be detected and filtered out.

The main aspect of the previous works on vision-based navigation that is superseded by our approach is the ability to adapt and maintain only one reference view for each place in the robot's map in response to environmental changes instead of keeping a history of multiple views to represent the same place over time.

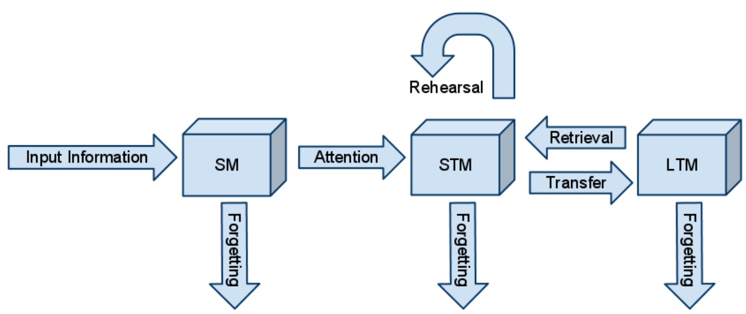

Fig. 1 Multi-store memory model. SM: Sensory memory. STM: Short-term memory. LTM: Long-term memory. Selective attention, which involves the LTM, determines what information moves from SM to STM. Through the process of rehearsal, information in STM can be transferred to LTM and be retained for longer periods of time. Information from the LTM store is retrieved using a process called "recall".

3 An Overview of the Memory Model

According to the basic model of Atkinson and Shiffrin [2], human memory is divided into separate stores for sensory memory (SM), short term memory (STM) and long term memory (LTM). 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. In our approach, perceptual attention includes detection of local image features for subsequent processing in the memory model. (see Fig. 1).

The concepts of the above memory model is used to update the the map, incrementally, by gradually adding information about new stable image 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 image features 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 (i.e. removes) unused feature points in LTM. LTM is used in turn by the attentional mechanism for selecting the new image features to update the map.

3.1 Map Representation

The robot's world is represented as a hybrid map consisting of two levels, global and local. Fig. 2 illustrates the hybrid map. On the global level, the world is represented as n optimized pose-graph. On the local level of the map, each node stores a spherical view representation of image features. The spherical views contain the 3D location of the image features on a sphere, so only the directions of the features (but not their distance or depth) from the centre of the sphere are stored. The centre of the sphere in this case corresponds to the centre of that node.

Each spherical view is initialised from an omnidirectional image recorded from the centre of each node in the global map. The spherical representation of the nodes creates a connection between the global and local levels of the map, where the group of image features is used as a qualitative descriptor for localisation on the global level, and the 3D location of these features on the sphere is used for estimating the heading needed for the navigation system at the local level [7].

Localisation on the global level is achieved by using an image similarity score based on the number of matched feature points between the current view of the robot and the group of points stored in each node. The robot localises itself to the node which has the highest similarity score in the map. Navigation on the local

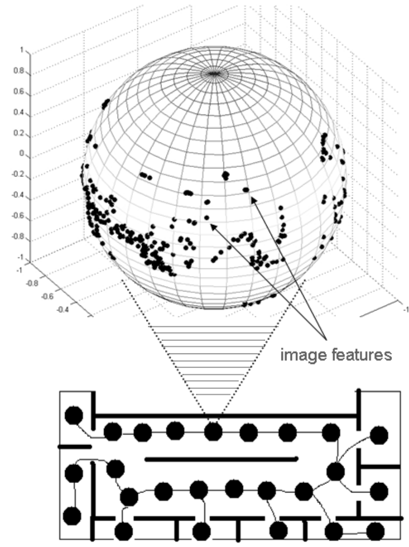

Fig. 2 The environment is represented as an adjacency graph of nodes on a topological level and each node on the metric level of the map represents the 3D location of image features on a unit sphere. Our method represents the direction of the features from the centre of the sphere, which corresponds to the centre of that node.

level is done by using multiple view geometry for spherical views to estimate the robot's heading during autonomous navigation. This navigation method is described in Section 4.3. The same image features used for navigation are also used to update the spherical views stored inside each node over time, in response to changes in the appearance of the environment.

4 Methodology

4.1 Map Updating Mechanism

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 stores information about features. The LTM represents the recent stable configuration of features in the environment and these are the 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.

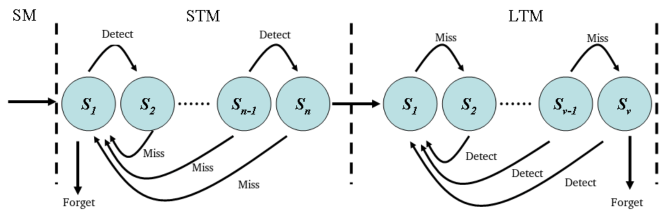

Fig. 3 The proposed multi-store memory model. SM: Sensory memory. STM: Short-term memory. LTM: Long-term memory.

Every time the robot finds the feature ("detect"), 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 a process of feature matching. In order to remain in the LTM, a feature has to be seen occasionally. 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 "detect" 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). We use multiple view geometry to transfer the image features from the current view to the spherical views of the nodes. The geometric method by which we update the spherical views of the map is presented in full detail in [7].

4.2 Temporal calibration for the updating mechanism

Temporal calibration means selecting the real-time unit in which the robot uses the memory system to update its map. In this work, it is assumed that the system is used by a mobile service robot working inside a house or public environment, where life is a series of daily episodes. This suggests that using days as a basic time unit would be a realistic choice. After each working day, during which it spends its time navigating inside the environment and visiting different nodes inside the map, the robot goes through what could be called a "sleep" period where it activates its memory system to update the map. However, in other situations where life and human activities do not follow daily cycles, the robot can adopt a time scale to update the map which reflects the natural cycle of activities in its surroundings.

Depending on the nature of the task, the robot may visit some nodes in the map multiple times during one day, whereas other nodes will be visited less frequently. This means that it is important to unify the rate at which the appearance of each

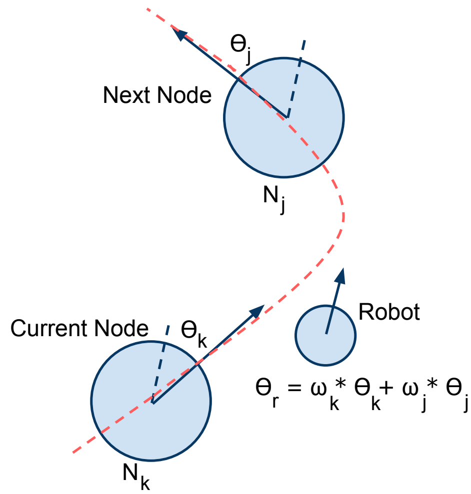

Fig. 4 The proposed navigation strategy using heading estimation. The robot is required to re-execute a path consisting of a number of key images which were recorded during a previous mapping stage. In the figure, N_k is the current node in the path and N_j is the next node. The red dashed line is the intended path. θ_k, θ_j are the relative orientations between the robot's heading and the reference orientation of the nodes N_k, N_j respectively. θ_r is the desired heading which results from a weighted sum of θ_k and θ_j.

node is updated. In order to achieve this aim, the robot selects, at the end of each working day and for each visited node in the map, one view only to use for map updating. The selected view is the one which has the highest number of matched points with the reference view of each visited node over the whole day.

4.3 Using the Map for Navigation

Every map can be judged by its usefulness for practical purposes. In our case the map is used for a daily path following routine inside a continually changing environment.

When robots work inside an indoor environment, their navigation generally is restricted to what the humans consider to be a path inside that environment, such as corridors and the areas between the furniture. These routes effectively simplify the task of navigation by limiting the robot to only one degree of freedom along the path. And by representing this path as a sequence of images, the following framework of the appearance-based approach for visual navigation is repeatedly used in the literature:

- The path is first built during a learning phase where the robot is controlled by a human operator. During this phase the robot captures a sequence of images along the path.
- A subset of the captured images is selected to represent the reference images along the path.
- During the replay phase, the robot starts near the first position and is required to repeat the same path.
- The robot extracts motion directions by comparing the currently observed image with the reference images along the path.

In this work we adopted a similar framework for visual path following using a sequence of nodes from the map. Fig. 4 illustrates the navigation strategy. First the robot localizes itself to one of the nodes in the path. This is done by selecting the node which has the highest similarity score with the currently observed view. Let S_k be the similarity score, i.e, the number of matched points. The similarity score is also computed between the current view and the next node in the sequence. Let S_j be the similarity score with the next node. After that the following ratio is computed:

$$\omega_k = \frac{S_k}{S_k + S_j},\ \omega_j = \frac{S_j}{S_k + S_j}. \tag{1}$$

Then the heading angle θ_r is computed as a weighted sum:

$$\theta_r = \omega_k * \theta_k + \omega_j * \theta_j. \tag{2}$$

where θ_k and θ_j are the relative orientation between the current view and the nodes N_k and N_j respectively (see Fig. 4). By following this navigation strategy, the nodes in the path can be considered as directional signs which lead the robot toward its goal.

In order to estimate the relative orientation between two views, such θ_k and θ_j in the above case, we use epipolar geometry to estimates the essential matrix $\mathbf{E}$, which is factored to give a rotation matrix $\mathbf{R} \in SO(3)$ and the skew-symmetric matrix $[\mathbf{t}]_\times$ of a translation vector $\mathbf{t} \in \mathbb{R}^3$ [11] as follows:

$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}. \tag{3}$$

After that, the the relative orientation is extracted from the rotation matrix $\mathbf{R}$.

5 Performance Evaluation

5.1 Map Adaptability

The main goal of the proposed memory model is to make the reference views of the map adapt to the changes in the appearance of the environment over time. In order to measure this adaptability we use the similarity between the views of the nodes and the reference views as a metric. Higher similarity means better representation for the environment where the similarity is measured as the number of matched feature points between two views. We compare how the similarity changes over time when the memory is used to update the reference views and again when the reference views were left static.

5.2 Map Consistency

Map consistency in our case means that the updating process of the reference views, which involves removing and adding image features over time, does not cause the map to degrade. If the map degraded over time, the robot would have difficulties to use the map for tasks such as autonomous visual navigation. Therefore, measuring the performance of executing a visual navigation task over time would be a good indicator of the quality of the map. In other words, the map is considered to be consistent if the performance does not drop over time.

In order to evaluate the performance of the proposed navigation strategy presented in Section 4.3, we use two metrics. The first is the length of the trajectory. If the length of the trajectory increased over time this would mean that the robot took more steps to execute the path due to poor directional information from the map. The second metric is the curvature of the executed trajectory by the robot [14]: the lower the curvature the smoother the trajectory. The smoothness of the trajectory is a good indicator of the consistency of the decision-action relationship in the navigation system. Similar to the first metric, if the curvature of the trajectory increased over time this would indicate that the robot performance is degrading.

Representing the trajectory of the robot as a curve in a 2D plane:

$$y = f(x), \tag{4}$$

the length of this trajectory can be calculated as:

$$L = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (f(x_{i+1}) - f(x_i))^2}, \tag{5}$$

where $(x_i, f(x_i))$, i=1...n, are the n points of the trajectory in Cartesian coordinates. The curvature of the trajectory at any point can be calculated as:

$$k(x_i) = \frac{f^{''}(x_i)}{[1 + (f^{'}(x_i))^2]^{\frac{3}{2}}}. \tag{6}$$

Using the above curvature factor, the smoothness of the trajectory can be measured as follows:

Fig. 5 An ActivMedia P3-AT robot equipped with an omnidirectional vision system.

$$B_E = \frac{1}{n}\sum_{i=1}^{n} k(x_i)^2, \tag{7}$$

where B_E is called the bending energy [16]. The bending energy can be understood as the energy needed to bend a rod to the desired shape. The less the energy the smoother the trajectory.

6 Experiment and Results

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. 5

The experiment was carried out inside the faculty office at the School of Computer Science in the University of Lincoln. We choose this area to conduct a long-term experiment because the room is designed as open offices where seven members of staff perform their daily activities. These activities result in changes to the appearance of the room over time. On the first day the robot was driven in a loop and a map with 10 nodes was created. For each node in the map, a spherical view of SURF features [3] was built. Using these spherical views, the map was used by the robot to perform a visual navigation routine from node number 1 to node 10. The 10 node route was repeated 38 times over a period of 8 weeks and after each run the robot used the memory model to update the map. Fig. 7 shows three images taken by the robot from the same place but at different times.

In this experiments the robot uses 3 stages in STM and 7 stages in LTM (one week) as the parameters for the memory. In other words, the robot rehearses new information for 3 days before transferring it to the LTM. In the LTM the robot forgets any information which has not been used for a week, taking into account the weekly episodic nature of our daily life.

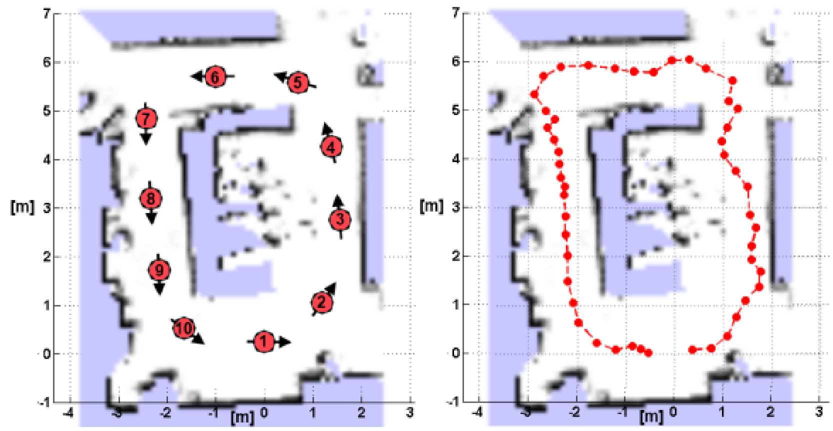

Fig. 6 Left:A laser based occupancy map for the area of the room where the experiments took place. Right: The trajectory of the path taken by the robot at one of the navigation episode during the experiment.

At the beginning of each run, the robot was placed in the vicinity of the first node. Then the robot performed global localization by matching the extracted feature points from its current view with all the reference views in the map and localized itself to the node with the highest number of matched points. Then the robot estimated its heading as described in Section 4.3. The obstacle avoidance procedure used in this work is as follows. When the robot receives a command to rotate, it checks its sonar range readings first. If the sonar ranges allow for movement, the robot simply executes the movement; otherwise, it turns 10^o in the opposite direction and then moves forward for a distance of 50 mm. After that it re-estimates the desired heading using the view from its new position. If both directions are blocked, the robot moves backward 100 mm and then re-estimates the desired heading from its new position. Finally, if the robot receives a command to move forward but there is no room for the movement based on sonar readings, the robot checks the sonar ranges on its right and left sides and then turns in the direction which has the most free space. This procedure is done in a recursive manner.

In order to obtain the ground truth data, we used Laser Range Finder (LRF) sensor with the GMapping library [9]. The GMapping algorithm provides a Simultaneous Localization and Mapping (SLAM) solution for static environments 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. This data provides us with information about the total distance travelled by the robot and the smoothness of the trajectory. Fig. 6 shows a laser-based occupancy map for the area of the room where the experiments took place.

The robot was able to perform the path following task successfully during all runs. As mentioned earlier, we use the similarity metric as an indicator for the adaptability of the map. The mean number of matched points between the view which has the best number of matching points and the reference views of the map was

Fig. 7 Three images recorded from the same place at different times. The appearance changes through the existence of new objects in the arena and the disappearance of others.

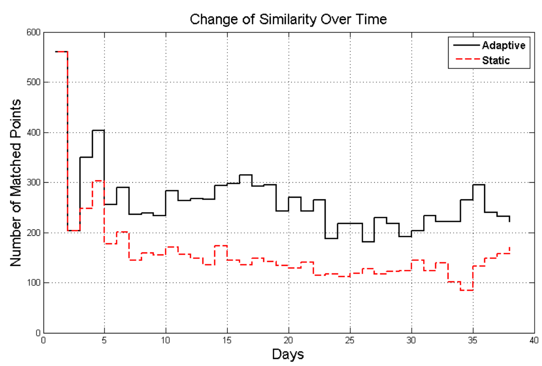

Fig. 8 A comparison between the static and the adaptive map showing the change of the similarity over the 38 runs for node number 4.

170.9 ± 84.6 when the static reference views were used for the map and 255.7 ± 92.6 when the adaptive map was used. This result shows the effect of using our memory model in increasing the similarity of the map to the environment. Fig. 8 shows how the similarity score changed over the 38 runs for node number 4 in the map.

The second metric is the change of the length of the trajectory over time. Fig. 9 shows that the length of the trajectory does not increase over time. The mean distance traveled over all runs was 19.9 ± 0.8 m.

The third metric used for the evaluation is the smoothness of the trajectory measured by the bending energy. Fig. 10 shows that the bending energy of the trajectory

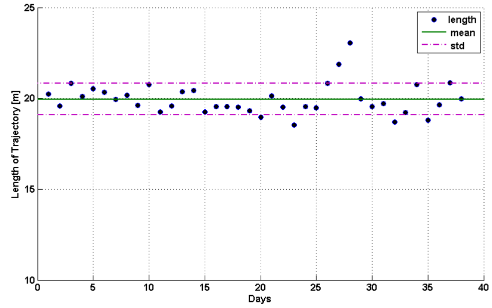

Fig. 9 The change of trajectory length over time. The mean distance travelled over all runs was 19.9 ± 0.8 *m*. Between days 27 and 28 a big box was delivered into the office, taking up part of the robot's path and forcing it to take a longer trajectory.

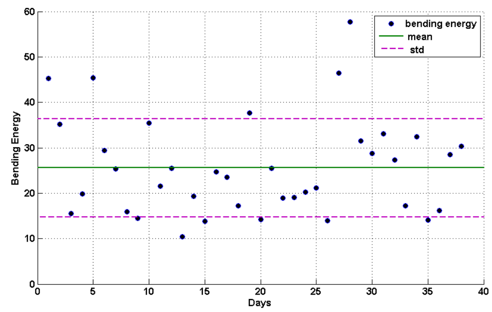

Fig. 10 The change of the smoothness of the trajectory measured by its bending energy. Between days 27 and 28 a big box was delivered into the office which affected the smoothness of the trajectory.

is not increasing over time but it is consistent. This means that the quality of the map is also consistent over time.

7 Conclusion

This paper presented am eight weeks episodic visual navigation experiment in a real office environment. An updating mechanism, based on short- and long-term memory concepts, incorporates a spherical view representation of image features, is used to keep robot's map up-to-date. The spherical views are used for navigation using multi-view geometry, as well as representing appearance signature of the environment. The results show that the proposed system has a persistent performance in such a real changing environments.

References

1. Andrade-Cetto, J., Sanfeliu, A.: Concurrent map building and localization in indoor dynamic environments. International Journal of Pattern Recognition and Artificial Intelligence **16**(3), 361–374 (2002)
2. Atkinson, R., Shiffrin, R.: Human memory: A proposed system and its control processes. The psychology of learning and motivation: Advances in research and theory **2**, 89–195 (1968)
3. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. Computer Vision and Image Understanding **110**, 346–359 (2008)
4. Bibby, C., Reid, I.: Simultaneous Localisation and Mapping in Dynamic Environments (SLAMIDE) with Reversible Data Association. In: Proc. Robotics: Science and Systems. Atlanta, GA, USA (2007)
5. Biber, P., Duckett, T.: Dynamic Maps for Long-Term Operation of Mobile Service Robots. In: Proc. Robotics: Science and Systems, pp. 17–24. Cambridge, MA, USA (2005)
6. Dayoub, F., Cielniak, G., Duckett, T.: (extended abstract) long-term experiment using an adaptive appearance-based map for visual navigation by mobile robots. In: Towards Autonomous Robotic Systems, pp. 400–401. Springer Berlin Heidelberg (2011)
7. Dayoub, F., Cielniak, G., Duckett, T.: Long-Term experiments with an adaptive spherical view representation for navigation in changing environments. Robotics and Autonomous Systems **5**, 285–295 (2011)
8. Dong, J., Wijesoma, S., Shacklock, A.: Extended rao-blackwellised genetic algorithmic filter SLAM in dynamic environment with raw sensor measurement. In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). San Diego, USA (2007)
9. Grisetti, G., Stachniss, C., Burgard, W.: Improved techniques for grid mapping with rao-blackwellized particle filters. IEEE Transactions on Robotics **23**(1), 34–46 (2007)
10. Hahnel, D., Triebel, R., Burgard, W., Thrun, S.: Map building with mobile robots in dynamic environments. In: Proc. IEEE International Conference on Robotics and Automation (ICRA), vol. 2, pp. 1557–1563. Taipei, Taiwan (2003)
11. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2 edn. Cambridge University Press (2004)
12. Liu, Z., Chen, W., Wang, Y., Wang, J.: Localizability estimation for mobile robots based on probabilistic grid map and its applications to localization. In: Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on, pp. 46–51 (Sept.). DOI 10.1109/MFI.2012.6343051
13. Migliore, D., Rigamonti, R., Marzorati, D., Matteucci, M., Sorrenti, D.G.: Use a single camera for simultaneous localization and mapping with mobile object tracking in dynamic environments. In: Proc. IEEE International Conference on Robotics and Automation (ICRA). Kobe, Japan (2009)
14. Muoz Ceballos, N., Valencia, J., Ospina, N.: Performance metrics for robot navigation. In: Electronics, Robotics and Automotive Mechanics Conference, 2007. (2007)
15. Ortega, S.: Towards visual localization, mapping and moving objects tracking by a mobile robot: a geometric and probabilistic approach. Ph.D. thesis, Institut National Polytechnique de Toulouse (2007)
16. Selekwa, M., Collins, E., Combey, J.: Multivalued Verus univalued Reactive Fuzzy Behavior Systems for Navigation Control of Autonomous Ground Vehicles. In: Proc. Florida Conference on the Recent Advances in Robotics (FCRAR), vol. 20 (2004)
17. Walcott-Bryant, A., Kaess, M., Johannsson, H., Leonard, J.: Dynamic pose graph slam: Long-term mapping in low dynamic environments. In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2012)
18. Wang, C.C., Thorpe, C., Thrun, S., Hebert, M., Durrant-Whyte, H.: Simultaneous localization, mapping and moving object tracking. The International Journal of Robotics Research **26**(9), 889 (2007)
19. Wolf, D.F., Sukhatme, G.S.: Mobile robot simultaneous localization and mapping in dynamic environments. Autonomous Robots **19**(1), 53–65 (2005)