An Adaptive Spherical View Representation for Navigation in Changing Environments

Feras Dayoub Tom Duckett Grzegorz Cielniak
Department of Computing and Informatics, University of Lincoln, Lincoln, UK

***Abstract*— Real-world environments such as houses and offices change over time, meaning that a mobile robot's map will become out of date. In previous work we introduced a method to update the reference views in a topological map so that a mobile robot could continue to localize itself in a changing environment using omni-directional vision. In this work we extend this long-term updating mechanism to incorporate a spherical metric representation of the observed visual features for each node in the topological map. Using multi-view geometry we are then able to estimate the heading of the robot, in order to enable navigation between the nodes of the map, and to simultaneously adapt the spherical view representation in response to environmental changes. The results demonstrate the persistent performance of the proposed system in a long-term experiment.**

***Index Terms*— Long-term SLAM, Persistent Mapping, Omni-directional Vision, Mobile Robot Navigation.**

I. Introduction

Maintaining an up to date representation of the surrounding environment is a necessity for mobile robots to be able to work with people in their everyday environment and to have the ability to localize and navigate using sensory information. Most work in mobile robot mapping considers only how to acquire the initial representation of the environment, but there has been very little work on how to update the map during long-term operation in changing environments.

An examination of the literature on visual mobile robot localization and mapping reveals two main approaches: metric methods [7, 18], which aim to estimate and track the absolute position of a robot inside a geometric map, and appearance-based topological methods [20, 19], which represent the environment as a graph where the nodes of this graph correspond to places in the real environment.

Between these two main branches, there is another approach which forms a hybrid between them [15]. In a hybrid map the environment is typically represented by a global topological map which connects local metric maps. The need for this hybrid type came from the complementary strengths and weaknesses of metric and topological methods. Full metric maps do not scale well to large-scale environments, while pure topological maps cannot be used for accurate navigation inside a node. However, the existing approaches (both topological and metric) require a certain level of stability (static world assumption). This leads to problems when applying these methods in our everyday environments where we tend to change the appearance of our surroundings by adding, removing or changing the arrangement of objects, which implies that the localization and mapping methods must have flexibility, robustness and adaptation ability along with some level of geometric accuracy.

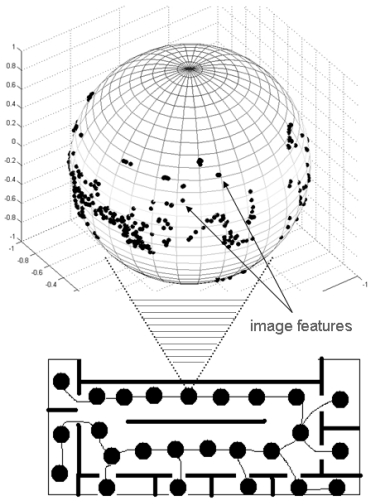

Fig. 1. Proposed Hybrid Metric-Topological Map. The environment is represented as an adjacency graph of nodes on a topological level and each node on the metric level of the map represents the 3D location of image features on a sphere. Our method represents the direction of the features (but not their distance or depth) from the centre of the sphere, which corresponds to the centre of that node.

In this paper, we propose a method to update the reference views of a hybrid metric-topological map for a changing environment, where 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, as shown in Fig. 1. This spherical representation of the nodes creates a connection between the topological level and the metric level of the map, by using the group of features as a qualitative descriptor for global localization on the topological level, and also using the 3D location of these features on the sphere for estimating the rotation angle needed for the navigation system at the metric level. In this way the proposed representation gives a balanced solution between the accuracy of geometric maps and the flexibility of topological maps.

The rest of this paper is structured as follows. Section II presents our initial work on an adaptive appearance-based map for long-term topological localization. Section III describes how to use the hybrid map for navigation. Section IV describes the method that updates the reference views at a geometric level. Section V presents the experiments and results obtained.

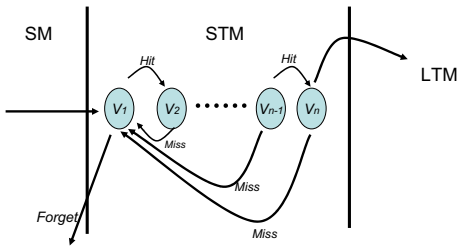

Fig. 2. The rehearsal stage in the STM shown as a finite state machine.

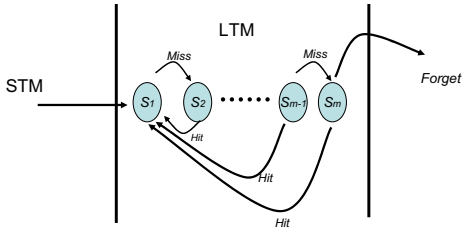

Fig. 3. The recall stage in the LTM shown as a finite state machine.

Finally we draw conclusions and discuss future work in section VI.

II. Background

In our previous work [6] we introduced a method to enable a mobile robot to update the reference views of a topological map for localization in a changing environment. 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).

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 robotic mapping, the sensory memory will contain image 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.

To initialise the map, the image data from the robot's first tour of the environment is used. In this work, the location and number of nodes in the map are selected by hand and assumed to remain fixed throughout the experiments. For each node, an image is recorded and local features are extracted using the SURF algorithm [4]. These features are used directly to initialise LTM, while STM for each node is initially assigned to be empty. Note that LTM represents the reference views of the map, which the robot uses to localize, so it must be initialized to contain the features from the first run, and STM is set to be empty, ready to be used in the rehearsal and recall stages for the subsequent runs.

Every time the robot visits an existing node (using an appearance-based global localization method to determine the current node), the processes of rehearsal and recall are carried out. Fig. 2 shows the rehearsal process for a stored feature in STM. In order to 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.

Fig. 3 shows the recall process for a stored feature in LTM. 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.

The question we address in this paper is the possibility to extend the above updating mechanism proposed for the topological level to the metric level of the map. In other words, is it possible to add and remove image features from the reference view during long-term operation and still maintain sufficient accuracy at the metric level to use the features for navigation between the nodes?

Different methods have been introduced to enable a mobile robot equipped with a visual topological map to navigate. Recently, Goedeme et al. [9] presented a system based on wide-baseline image features matching which enables the robot to follow a pre-recorded sequence of omnidirectional images. Guerrero et al. [10] presented a hierarchal localization system which uses 1D three view geometry to achieve local metric localization which can be used to navigate the robot. Booij et al. [5] built their navigation system based on heading

estimation to achieve a hill-climbing behaviour with no need for a pre-recorded path to follow.

III. Heading estimation using the spherical views

In many tasks, the robot needs not only to find its current location, but also to use the sensory information along with the map to navigate. In this section we describe the navigation scheme we use.

Our approach does not store a global metric map of the environment. Instead we estimate the heading of the robot relative to the current node, by estimating the relative orientation of the current view of the robot with respect to the stored reference view. In this way, the reference views function as way-points that the robot can use to travel from one place to the next.

In our case, the desired heading is estimated using the epipolar geometry for spherical cameras [1]. The model of the spherical camera consists of a unit sphere whose centre is the centre of the curved mirror. The omnidirectional camera we are using can be treated as a spherical camera because it is a central omnidirectional camera. A central omnidirectional camera has a point called an optical centre, on which all projection rays meet. So after calibrating the camera using the toolbox by Scaramuzza et al. [17], each reference image is represented by a spherical view (Fig. 1) where the features points are projected on a unit sphere.

Given two spherical views with centres C and $C^{'}$, a scene point P can be back-projected through the two spheres to the centre of projection for each camera. Let X represent the position of P in the reference frame of C and x represent the projection of X on the sphere, then we can write:

$$\lambda x = X, \quad \lambda \in \mathbb{R}_{+} \tag{1}$$

In the same way for the second camera we will have:

$$\lambda' x' = X', \quad \lambda' \in \mathbb{R}_{+} \tag{2}$$

where x', X' are the 3D points in the frame of $C^{'}$. Assuming that $C = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$ with $\mathbf{R}$ and $\mathbf{T}$ expressing the transform of the camera coordinates between C and $C^{'}$, we can write:

$$X' = \mathbf{R}X + \mathbf{T}, \tag{3}$$

then by substituting Eqs. 1 and 2 into Eq. 3, we get:

$$\lambda' x' = \lambda \mathbf{R} x + \mathbf{T}. \tag{4}$$

which leads to the following relation based on the epipolar geometry for spherical cameras:

$$(x')^T \mathbf{E} x = 0, \tag{5}$$

where $\mathbf{E}$ is the essential matrix. This matrix can be linearly solved using eight pairs (or more) of corresponding points from the two spheres [16]. 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. Due to that and the fact that the false matches will always be part of the matching process, using the RANSAC algorithm [8] is a very efficient way to minimize the effect of the outliers and find the best essential matrix to fit most of the points.

The robot in our case is working on a planar floor which means that the rotation between the spherical views will only be around the vertical axis. Using this fact, we can simplify the process of estimating the essential matrix by restricting it to the following sparse form [14], assuming translation in x-y plane and rotation around z-axis:

$$\mathbf{E} = \begin{bmatrix} 0 & 0 & e_{13} \\ 0 & 0 & e_{23} \\ e_{31} & e_{32} & 0 \end{bmatrix}. \tag{6}$$

Based on the method introduced by Hartley and Zisserman in [11], the essential matrix is factored to give Eq. 7 which contains the rotation matrix $\mathbf{R} \in SO(3)$ and the skew-symmetric matrix $[\mathbf{T}]_\times$ of the translation vector $\mathbf{T} \in \mathbb{R}^3$:

$$\mathbf{E} = [\mathbf{T}]_\times \mathbf{R}. \tag{7}$$

This will generate four possible combinations of $\mathbf{T}$ and $\mathbf{R}$. By choosing the combination which makes the reconstructed points lie outside the sphere we can determine the correct one (applying the positive depth constraint).

After the estimation of $\mathbf{T}$ and $\mathbf{R}$, the robot can find the heading toward the reference view, α, using the translation direction $\mathbf{T}$:

$$\alpha = atan(\mathbf{T}[y], \mathbf{T}[x]), \tag{8}$$

and also the rotation between the current view and the reference view, γ, using the rotation matrix:

$$\mathbf{R} = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{9}$$

IV. Updating the spherical views

Updating the reference views of the map based on the proposed memory model means removing old unused features and adding new features during long-term operation of the robot. So in order to preserve the ability to use the updated spherical views for heading estimation, the feature points which need to be moved to the STM and LTM stores of each node should be located on the reference sphere as if these features were seen from the same point where the node was first created. This ensures that the robot will keep the ability to use the reference views for heading estimation and therefore navigate using the map.

In order to achieve this, we reconstruct the 3D position of feature points shared between one view from the current visit and one view from the previous visit to the node. The current and previous views are each obtained by selecting the image in the recorded sequence for that visit with the highest similarity score to the reference view. The 3D position of the shared points can 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 these new features to the STM into their correct position on the sphere we use a simplified version

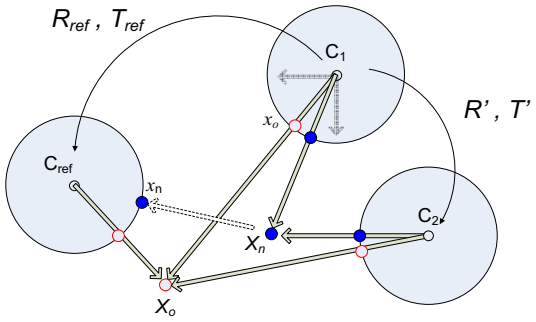

Fig. 4. Reference view updating. The current view C_1 is matched with the previous view C_2 and the reference view C_{ref} to estimate the coordinates of new features in the spherical representation of the reference view.

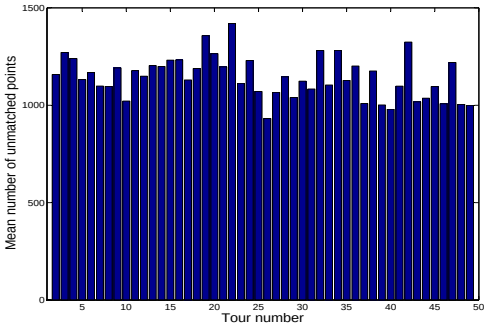

Fig. 5. The average number of unmatched points between consecutive tours as a measure for the changing appearance of the environment.

of what is known in the computer vision literature as multi-baseline stereo [13]. In our case, we only use two stereo pairs between three views: the reference view and the last two views of 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 [11]

In Fig. 4, let X_o be one of the reconstructed positions for an image feature which is shared between the three views (C_1, C_2, C_{ref}) where C_{ref} is the reference view of the node and C_1, C_2 are the views from the current visit and the previous one, respectively.

Based on the stereo views (C_1, C_2), we can write:

$$X_o^{C_2} = \lambda_2 x_o, \tag{10}$$

where λ_2 is the depth of X_o based on the unknown scale of the stereo views (C_1, C_2), $X_o^{C_2}$ is the representation of X_o in the reference frame of C_1 and x_o is the projection of $X_o^{C_2}$ on the unit sphere of C_1.

Also, in the reference frame of the view C_1 and based on the stereo views (C_1, C_{ref}), we can write:

$$X_o^{ref} = \lambda_{ref} x_o, \tag{11}$$

where λ_{ref} is the depth of X_o based on the unknown scale of the stereo pair (C_1, C_{ref}) and X_o^{ref} is the representation of X_o in the reference frame of C_1.

Eqs. 10 and 11 mean that a point X_o shared between the three views will have different values $(X_o^{ref}, X_o^{C_2})$ depending on the scale of the reconstruction. This also means that we can convert between the different unknown scales:

$$X_o^{ref} = s X_o^{C_2}, \quad s \in R. \tag{12}$$

The value of s is estimated such that it minimizes the distance error between the 3-D point's correspondences between the two stereo pairs (C_1, C_2) and (C_1, C_{ref}). Outliers are rejected using robust statistics [3].

Now, let X_n be a reconstructed position of an image feature shared between the two views (C_1, C_2) but which does not exist in the view C_{ref}. In order to find the projection of X_n on the sphere of C_{ref}, first we need to convert to the scale of the stereo pair (C_1, C_{ref}) using s:

$$X_n^{ref} = s X_n^{C_2}, \tag{13}$$

where $X_n^{C_2}$ is the representation of X_n in the reference frame of C_1 based on the scale of the stereo views (C_1, C_2) and X_n^{ref} is the representation of X_n in the reference frame of C_1 based on the scale of the stereo views (C_1, C_{ref}).

Then, as shown in Fig. 4, the view C_1 and the reference view C_{ref} are related by a rigid body displacement represented by the rotation matrix $\mathbf{R}_{ref} \in SO(3)$ and the translation $\mathbf{T}_{ref} \in \mathbb{R}^3$. We can transform X_n^{ref} to the frame of the reference view C_{ref} as follows:

$$X_n^{C_1} = \mathbf{R}_{ref} X_n^{ref} + \mathbf{T}_{ref}. \tag{14}$$

Finally, the position of the new feature in the STM store of the reference view sphere, x_n, can be found by normalization:

$$x_n = \frac{X_n^{C_1}}{\|X_n^{C_1}\|}. \tag{15}$$

V. RESULTS

The following experiment was designed to evaluate the ability of the system to add and remove image features from reference views of the map while at the same time still being able to use the features for heading estimation.

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. 8). For local feature extraction we use the SURF algorithm [4]. The experiment was carried out in our robotics lab where we collected 1385 images by driving the robot in a loop. The images were generated from 50 tours over a period of 3 days. After each tour the appearance of the lab was changed manually. The changes involved 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. 9 shows two images taken from the same node at different times.

To characterize the dynamics of the test environment, we chose the following metric: the number of unmatched features

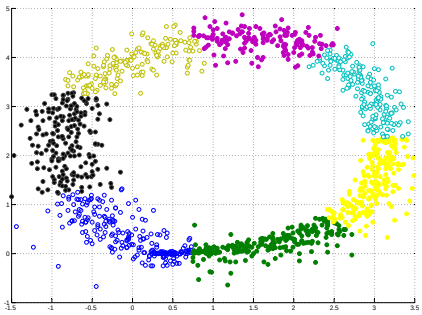

Fig. 6. Ground truth positions of the recorded images obtained from the laser-corrected odometry. The constructed map consists of seven nodes, each colour representing the group of images which belong to the same node.

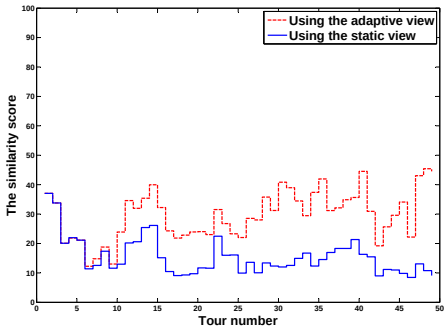

Fig. 7. The similarity score for node 4 using the static view and the adaptive view.

between omnidirectional images for the same location over a period of time. The number of unmatched features is a good measure for the changing appearance because it captures the features from the old appearance which do not exist in the new images of the location and also the features of the new appearance which did not exist in the old images. If the environment is static this number should be constant but in a changing environment this number will change considerably.

For each consecutive tour, we find the average number of unmatched points over all the nodes in the map. Fig. 5 shows how this number changes over time and we can see that between tours 24 and 25 there was no great change but between tours 20 and 21 the environment experienced a big change.

In order to obtain the ground truth positions for the collected data, the starting points for all 50 tours were initialized from a fixed point inside the room while each image was recorded along with a laser scan and odometry. This enable us to use LODO [12], a library for laser-based correction of odometry, attaching each image with a 2D position and a rotation relative to the starting point. The first tour is used to create the topological map which consists of 7 nodes (selected manually). Fig. 6 shows the ground truth positions of the images, each colour representing the group of images which belong to the same node. The rest of the image sequence is used as input for

Fig. 8. The experimental platform. An ActivMedia P3-AT robot equipped with an omnidirectional vision system.

the localization system. We used global localisation based on place recognition using a similarity score between the current and the reference views (winner-takes-all) as a first stage to locate the robot in one of the nodes [6].

To find the similarity score between two groups of feature points, we use the number of corresponding features M_{ij} between the two groups based on a nearest neighbour (NN) matching scheme using the value 0.7 as a threshold between the nearest and second-nearest neighbour, following [4]. The similarity score between view V_j and a reference view V_i can be defined as:

$$S_{ij} = \frac{M_{ij}}{K_i} * 100 \qquad (16)$$

where K_i is the number of features in the reference view V_i.

During a visit to a node in the map, the robot will capture a number of images as it goes through the node. Among these images the image with the highest similarity score is used to represent the view of the node for that visit and it is then used to update the reference view using the proposed updating mechanism.

After the global localization step, the reference view of the node and the input image is used to estimate the rotation between the two views using Eq. 9, and then the estimated rotation is compared with the ground truth obtained from the laser-stabilized odometry. Using the sequence of collected images we repeat the same experiment once using static reference views for the map and then using the adaptive views. The static reference views are created from the first run (similarly the first run is used to intialise LTM as described in Section II) and the subsequent runs are used for localization.

Table. I shows a comparison using several performance measures, showing mean and standard deviation, between the

TABLE I
LONG-TERM LOCALIZATION RESULTS

Measure	Comparison measures	
	Static Map	Adaptive Map
Error in estimating the rotation	$4.2^o \pm 4.1^o$	$4.5^o \pm 4.6^o$
Mean number of matched points	81.8 ± 43.8	118.3 ± 54.4
Number of matched points > 50	77.0%	95.1%

Fig. 9. Two panoramic views from the same place at different times.

two cases using 8 stages for the LTM and 3 stages for the STM. As shown in the first row, the error in the estimation of the rotation did not drop significantly whereas the average number of the winning matched points, which is used for the global localization, has increased noticeably when we used the adaptive views, as shown in the second row. As the environment changes over time, the winning number of matched points became smaller when the static reference views were used. As shown in the third row, during global localization the winning number of matched points was over 50 in 77.0% of cases for the static map and 95.1% for the dynamic map.

Fig. 7 shows how the similarity score changed over the 49 tours for node number 4 . As shown, using the adaptive views gave a higher similarity score, while for the static view the similarity score sometimes dropped below 10% as in tour 17.

VI. Conclusions

This paper introduced a method to enable a mobile robot working in a non-static environment to update an internal representation of its environment in response to the changes in the appearance of that environment. We extended our previous work on long-term mapping for a topological map [6], by adding a metric level where each node represents the 3D location of the corresponding image features on a sphere. The updating mechanism is based on long-term and short-term memory concepts which use local image features to update reference views in a hybrid metric-topological map, while preserving the ability to use the updated reference views for heading estimation based on multi-view geometry of spherical cameras.

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. Bigger hybrid metric-topological maps will also be built and further tests on localization and navigation will be carried out. 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.

References

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

[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] E. Bandari, N. Goldstein, I. Nesnas, M. Bajracharya, and N. RIACS. Efficient calculation of absolute orientation with outlier rejection. *BMVA Syposium on Spatiotemporal Image Processing*, 2004.

[4] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features. In *Proc. European Conference on Computer Vision (ECCV)*, 2006.

[5] O. Booij, B. Terwijn, Z. Zivkovic, and B. Krose. Navigation using an appearance based topological map. *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007.

[6] F. Dayoub and T. Duckett. An adaptive appearance-based map for long-term topological localization of mobile robots. In *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[7] P. Elinas, R. Sim, and J. J. Little. sigmaSLAM: Stereo vision SLAM using the Rao-Blackwellised particle filter and a novel mixture proposal distribution. *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, page 1564–1570, 2006.

[8] 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*, 24:381–395, 1981.

[9] T. Goedemé, M. Nuttin, T. Tuytelaars, and L. Van Gool. Omnidirectional Vision Based Topological Navigation. *International Journal of Computer Vision*, 74(3):219–236, 2007.

[10] J. J. Guerrero, A. C. Murillo, and C. Sags. Localization and matching using the planar trifocal tensor with bearing-only data. *IEEE Transactions on Robotics*, 24:494–501, 2008.

[11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, March 2004.

[12] A. Howard. Laser-stabilized odometry (lodo_driver). Available from http://robotics.usc.edu/˜ahoward.

[13] S. B. Kang and R. Szeliski. 3-D scene data recovery using omnidirectional multibaseline stereo. *International Journal of Computer Vision*, 25(2):167–183, November 1997.

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

[15] B. Kuipers and Y. T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Toward Learning Robots. MIT Press, Cambridge, Massachusetts*, page 47–63, 1993.

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

[17] D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easily calibrating omnidirectional cameras. *Proc. of the IEEE International Conference on Intelligent Systems, IROS06, Beijing, China*, 2006.

[18] S. Se, D. Lowe, and J. Little. Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks. *The International Journal of Robotics Research*, 21(8):735, 2002.

[19] C. Valgren, A. Lilienthal, and T. Duckett. Incremental Topological Mapping Using Omnidirectional Vision. In *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2006.

[20] Z. Zivkovic, O. Booij, and B. Kröse. From images to rooms. *Robotics and Autonomous Systems*, 55(5):411–418, 2007.