Vision-based Topological Map Building and Localisation using Persistent Features

Deon G. Sabatta
Mobile Intelligent Autonomous Systems,
Council for Scientific and Industrial Research
Pretoria, South Africa
dsabatta@csir.co.za

***Abstract*— This paper proposes a topological mapping technique that utilises persistent SIFT features to reduce the amount of data storage required. It delivers, as an output, a topological map that lends itself well to conventional path planning techniques. The approach assimilates features into a statistical model which promises improved data association. Experiments were performed using omnidirectional camera images from the Cogniron dataset. A map was constructed from one of the supplied routes and the performance of the localisation algorithm evaluated using another route within the same environment. Thereafter the map was updated using the comparison route and the results discussed. The statistical feature association approach is shown to be more robust than conventional methods.**

I. Introduction

Most problems in autonomous robotics require systems that develop an environmental map and then localise themselves within this map. Maps may typically be topological [1], [2], metric [3], [4] and more recently a hybrid metric-topological model [5], [6]. While metric maps have received much attention, their accuracy and completeness are often not required for tasks such as navigation and path planning. This accuracy is also accompanied by a high cost in terms of memory and processing time required to create them. Further work investigating the consistency of metric maps by Castellanos, [7], shows that these maps may become statistically inconsistent unless loop closure is performed frequently. This requirement often cannot be met. In contrast, topological maps store only unique or distinct areas and the relationships between them. This is similar to the way humans store information of their environments. Hybrid metric-topological models combine both these methods by storing metric submaps associated with topological landmarks. This affords the accuracy of metric map localisation without the problems associated with statistical inconsistency.

Our approach is based on the topological mapping paradigm using local image features extracted from a panoramic image of the environment. These features are collected into groups based on their persistence within the local environment. In the construction of the topological map, each region always shares some features of nearby environments. Navigation through this map may then be accomplished by following a trail of common features between the environments en route to the final destination. Through careful selection of features, the amount of data required to define an environment is reduced and rapid localisation without a priori information is possible. The proposed algorithm is thus ideally suited to solving the kidnapped robot problem.

Fig. 1. Typical panoramic image showing all the SIFT features in red with the persistent SIFT features highlighted in yellow.

The remainder of this paper is structured as follows. In the next section we discuss related work in the field of topological mapping as well as the improvements of the proposed algorithm. In Section III we introduce the methods of map building and localisation employed by our algorithm. In Section IV we provide the experimental results and in Section V we present our conclusions and possible future work.

II. Related Work

The concept of topological mapping was introduced into the field of robotics following studies of human cognitive mapping undertaken by Kuipers [8]. Since then, much progress has been made in the field of vision-based topological mapping. Topological mapping lends itself well to vision based approaches as exact localisation is not required. This allows the use of image properties that are not specific to an exact location and also permits some variability in the environment. Ulrich and Nourbakhsh [9] classified regions through the use of global image features using a colour image histogram. In the work of Tapus and Siegwart [2], topological regions are classified according to a “fingerprint”. This fingerprint is a combined descriptor of an environment formed using vertical edges extracted from panoramic images, colour regions and corners extracted from laser range data.

Topological maps have also been successfully built using local features in images. Rybski et al. [10] use the Kanade-Lucas-Tomasi (KLT) feature tracking algorithm to build a topological map incrementally using panoramic images. Booij et al. [11] have developed an appearance based topological mapping algorithm using Scale Invariant Feature

Transform (SIFT) features. SIFT features provide descriptors that are invariant to scale and rotation changes and robust to photometric changes, enabling topological mapping based on image similarity. Our method is similar in that topological regions are defined as a collection of SIFT features, however we achieve a reduction in the number of stored features by only considering persistent features that exist in the vicinity of a topological region. Figure 1 shows a typical panoramic image with the conventional SIFT features indicated in red. The highlighted SIFT features in yellow are those which are reliably detected and visible within a domain around the current frame. Furthermore, we also introduce a statistical framework for the association of stored features with the currently visible features improving the matching reliability.

Topological mapping has also been investigated by Fraundorfer et al. [12] as an application of a content based image retrieval system by storing all observed images of an environment in a database. Localisation is then achieved by attempting to match the current image with the images stored in the database.

III. METHOD

This section describes our method used to create topological maps of the environment. We start by extracting interest points together with the associated descriptors from the omnidirectional input images. These features are then tracked from image to image using a multi-image association strategy. Once a feature has persisted for long enough in the environment it is added to the current environment's feature list. When less than a certain number of features are still visible, a new node in the topological map is created and linked with the previous node.

Localisation is performed by matching features in the current image with those stored in the topological map. Feature matching is performed using the Mahalanobis distance with outlier compensation to approximate the closeness of the match. A weighted matching function is then used as a measure of certainty that the currently observed surroundings match a previously stored environment.

A. Feature Extraction and Association

The omnidirectional camera images are obtained from a hyperbolic mirror mounted on top of a camera. These images are then warped into 800 x 200 pixel panoramic images for processing. Features are extracted using a variant of the SIFT algorithm [13] which identifies blobs in the image by finding the maxima within a Gaussian scale space. One problem with this method is that large blobs near the edge of the image are not reliably identified due to inconsistencies in the definition of a Gaussian blur on partial data. Since we have a panoramic image that spans a full revolution we copy portions of the left side of the image onto the right and vice versa. We copy a segment equal to the height of the panoramic image from each side resulting in a 1200 x 200 pixel image. The SIFT algorithm that we use has been modified to incorporate colour information. The colour panoramic images are converted to the CIE *Lab* colour space. The *Lab* colour space was intended to be used as a perceptually uniform, luminance invariant colour space where the *L* component represents the luminance of the image and the *a* and *b* components are chrominance components representing the colour information. In our system, the SIFT detector operates on the luminance component of the image, while the feature descriptors are taken from the *a* and *b* colour components, at the location of the interest point, resulting in a 256 dimension feature descriptor. In this manner, colour information is also included in the feature descriptor resulting in improved descriptor matching in some environments at the cost of more feature terms. While the *Lab* colour space conversion is reasonably costly in terms of implementation, the conversion may be parallelised in hardware to improve the performance of the feature extractor.

In some topological mapping algorithms [14], [15], these interest points, together with their descriptors, are used to match similar looking images as a form of topological localisation. The problem with these methods is that from a single frame, we cannot be sure of the stability of the feature point or its descriptor. In [16] it is shown that descriptors may vary by an order of magnitude more over large changes in view-point than in adjacent frames. In our algorithm, we track the evolution of a feature over several input frames, placing us in a position to discard unstable features as well as provide a statistical representation of the descriptor as we move through the environment.

To accomplish this we could simply match features from adjacent frames and record the descriptors. In some cases, however, a feature may be absent from a single frame due to occlusion, sensor noise or some other change in the environment, such as lighting. Clearly, these largely stable features should also be included in the feature set that describes an environment.

To create these associations, the extracted features are matched between successive frames with a look-ahead match to track features that may skip a frame. To match the features we follow the approach of [13]. Given two images A and B, with feature descriptor sets F_A and F_B, a feature $f_A^n \in F_A$ is matched to the feature $f_B^m \in F_B$ if the Euclidean distance $d(f_A^n, f_B^m)$ between the two feature descriptors satisfies the constraint,

$$d(f_A^n, f_B^m) \times T < d(f_A^n, f_B^p) \quad \forall \quad f_B^p \in F_B, \qquad (1)$$

where T is some threshold value used in the matching process. This form of matching, while reasonably robust, does permit multiple matches to a single feature in the second image. To overcome this problem, matches are found from image A to B as well as from image B to A, and only matches present in both sets are taken as true matches.

To perform the look-ahead matching, we process three consecutive images, A, B and C for matches between them. Using these matches we establish a $n \times 3$ matrix of matches between descriptor IDs, where matching from A through B to C is preferred over matching between A and C. This matching matrix is then passed onto the map building algorithm to produce the topological map.

B. Environment Definition

To better describe an environment, we introduce the notion of a feature *"snake"*. A snake of features is a list of feature IDs related to successive frames from the dataset for which a positive match was obtained using the method of Section III-A. The heart of the map building algorithm is centred around a dynamic 2D array of these *"snakes"*. Each row within this array represents a single feature that is being tracked. Each column represents the frames in which the features are present. To prevent an explosion in the size of the array it is constantly pruned to remove old features on the top and empty columns on the left while new features are being added to the bottom and new frames on the right. For this reason it is also useful to maintain an incrementing list of unique landmark IDs and frame numbers for reference. When merging the match array from the previous section, direct matches between adjacent frames are preferred over matches between features that skip a frame.

When a snake of features fails to have a matched feature for two successive frames, no further feature matches will be possible on this snake as only a single look-ahead frame is employed. When this is found, the snake is removed from the array and added to the environment information. The length of the snake is important for the performance of the algorithm. If the snake has too few features, the statistical descriptor will not be well formed and lead to poor matching results. For this reason we put a limit on the minimum length as well as the minimum number of non-zero descriptors.

An environment within the topological map is then defined as a collection of landmarks or feature snakes visible in the region of the keyframe – which also describes the location of the environment. To reduce the amount of data stored for each environment, we construct a second-moment statistical descriptor for each feature snake by calculating the mean $\bar{\mu}_i \in \mathbb{R}^{256}$ and the covariance matrix $S_i \in \mathbb{R}^{256\times256}$. To simplify this definition, we assume that the elements of the feature descriptor are independent, thus reducing S_i to a diagonal matrix.

In addition to the landmark descriptors we also store the relative angular location of each of these features with respect to the robots current heading in the keyframe and the relative location of the current environment to the previous one. This information may later be used for the purposes of localisation and ultimately navigation.

C. Localisation

Localisation is performed by matching currently visible features extracted from the environment with the database of features stored for each region. This matching may either be performed with a single feature extracted from a panoramic image or from a statistical descriptor of another landmark for the purposes of loop closing during map building.

The metric used to calculate the match is the Mahalanobis distance given by,

$$d_i^2 = (\mathbf{x} - \mu_i)^{\mathrm{T}} S_i^{-1} (\mathbf{x} - \mu_i),$$

for a feature, $\mathbf{x}$, and by,

$$d_{ij}^2 = (\mu_i - \mu_j)^{\mathrm{T}} (S_i + S_j)^{-1} (\mu_i - \mu_j),$$

when matching two statistical descriptors. To account for outliers, when calculating this distance we ignore the contributions from the largest 10% of normalised distance elements in the above sum.

Using this metric we may determine the number of matches between two environments by accepting any match less than a certain threshold, χ^2. When multiple landmarks within the current or stored environments match a single landmark, the landmark pair with the lowest distance is accepted as the correct match.

Given the number of matching landmarks, n, between any two regions and the mean Mahalanobis distance of these matches, $\bar{d}$, we construct a weighted evaluation function, with threshold w_t, that determines if a match exists between any two regions.

$$w = n + k_1 \left(k_2 - \bar{d}\right)^3 \geq w_t \tag{2}$$

Without the second term in the above equation, we would consider two environments to match one another when a certain number of landmarks are found to be common between them. The inclusion of the second term in the above function increases the certainty of a match as the average Mahalanobis distance decreases below some value k_2. This allows us to accept either a few very good matches or many poor matches as a positive environment association. Ideally a more smooth threshold function with the same shape as x^3, such as $\mathrm{atanh}(x)$, would be preferred although we have chosen the simpler function to speed up the implementation.

Once a region has been determined, an approximation of the relative orientation of the robot with respect to the original orientation of the region may be determined by calculating the mean angular displacement of the matched features.

D. Map Building

For the purposes of topological map building, a map is composed of a set of environments or distinct regions within the world. Naturally, there must also be a way to identify when the robot reencounters these environments as well as a method of defining which of these environments are in relative proximity to one another. With the definition of an environment, and localisation within these environments, discussed above, the problem of map building now reduces to that of when to define a new region and how to merge current information with an existing region. To do this, we process the input frames in sequence as they are obtained from the omnidirectional camera. The amount of work required for map building may be greatly reduced by only considering every k^{th} input frame (however, every input frame is used for snake generation).

For each frame under consideration, we collect the set of all visible, persistent landmarks in an interval of f_n frames

around the current frame[1]. For the case of the first frame, f_n frames from the start, we simply add the first region to the environment list together with the orientation of the features with respect to the robot's heading.

For each subsequent frame, we attempt to match the visible landmarks with those in the environment database according to (2). From here, three possible conditions exist:

1) The current environment obtains the best match with the currently active region.
2) The current environment obtains the best match with another region.
3) The current environment does not match any of the stored regions.

For case 1 we simply move on to the next frame and repeat the process. For case 2, we set the region with the best match to be the current region, update the connectivity matrix to indicate a connection between the previous and current regions and merge the landmark sets. For case 3 we add the current frame to the region database and update the connectivity matrix to indicate a connection between the previous and new regions.

Occasionally we may encounter a frame where very few persistent landmarks are visible. This presents a problem to the map building algorithm as there may not be sufficient features to allow a positive match to future frames. Two options exist at this point, either ignore the frame and continue until a more suitable frame is found or increase the interval size f_n until a suitable number of features are visible. We have found that a joint approach works best, where the interval is expanded to some limit and then if sufficient features are still not visible, the frame is discarded.

Merging of landmarks: When an environment is revisited, the currently visible landmarks can be merged with those in the environment. This allows us to improve the quality of the existing landmarks as well as to include additional landmarks that may not have been initially incorporated into the database. If a landmark matches one that is stored in the environment, we merge the two landmarks, otherwise we add it as a new landmark.

We merge two landmarks using a maximum-likelihood estimate of the new parameters as,

$$S_T = (S_1^{-1} + S_2^{-1})^{-1} \quad (3)$$

$$\mu_T = S_T(S_1^{-1}\mu_1 + S_2^{-1}\mu_2). \quad (4)$$

When merging the currently visible frame with one in the database, we do not update the relative position of the region or the relative angular location of the existing landmarks. The new landmarks are assigned an angular position derived from the assumed orientation of the robot based on the procedure described in Section III-C.

IV. Experimental Results

In this section we investigate the performance of our method to perform topological localisation and map building using the Cognitron dataset available from [17]. This dataset provides several runs through a typical household environment with changes in the environment and the presence of dynamic objects (people). We use this dataset to test the performance of our algorithm in the presence of occlusion and dynamic changes while providing a basis for comparison with other algorithms. The experimental results will use the 1^{st} and 4^{th} runs from *Home1*. *Run 1* comprises a single trip around the environment with no dynamic objects or changes in the environment, while *Run 4* provides several smaller loops in the environment in the presence of people and with changes in the environment from *Run 1*.

For the SIFT feature extraction, we use a detection threshold of 0.01 and an edge threshold of 4. The threshold used for SIFT matching in (1) is $T = 1.5$. We only consider feature snakes with a length of 30 features or more to ensure that the statistical model of the descriptor is well formed. When determining if two features are considered a positive match, we use a threshold of $\chi^2 = 40$, taken from a χ^2 table. For the constants in equation (2) we use $k_1 = 0$ and $w_t = 7$ for map building and $k_1 = 0.025$, $k_2 = \sqrt{40}$ and $w_t = 6$ for localisation. The values of k_1 and k_2 are chosen to weight the second term in (2) so that no additional contribution is present if the average distance matches the threshold value and that a single match of distance 0 will satisfy (2) during localisation. The choice of w_t influences the average distance between nodes of the topological map and was chosen experimentally. We ignore the second term of (2) during map building as matches between the same landmarks within the stored region and the currently visible frame result in a Mahalanobis distance of 0 affecting the performance of (2). For the case of localisation, an exact match, such as the one described above, is virtually impossible due to noise. During map building we only consider every 3^{rd} frame, use an initial value of $f_n = 3$ and increase it to a maximum of 15 in an attempt to increase the number of visible features.

A. Localisation

To test the performance of the localisation algorithm, we first construct a map of the environment using one dataset. Thereafter we attempt to localise the position of the robot from the second dataset within the map generated from the first. By virtue of the method used to construct the map, the dataset used to construct the map always has a 100% correspondence to the regions within the map.

When localising the position of the robot in *Run 4* with a map generated using *Run 1*, we obtain a 78.5% match to a nearby region within the map. The algorithm may not always localise to the nearest region as passing through "doorways" changes the visible set of features, and the match will then be to the nearest region containing the currently visible features. The proposed algorithm presents an improvement over similar algorithms [15] where only features descriptors from single frames are considered. The 78.5% match obtained is reasonable considering that the robot traverses regions in *Run 4* that are not visited in *Run 1*. For these regions, a suitable match cannot be expected.

[1]Causality requires that we only consider frames occurring before the current frame unless the map is being built off-line from collected data.

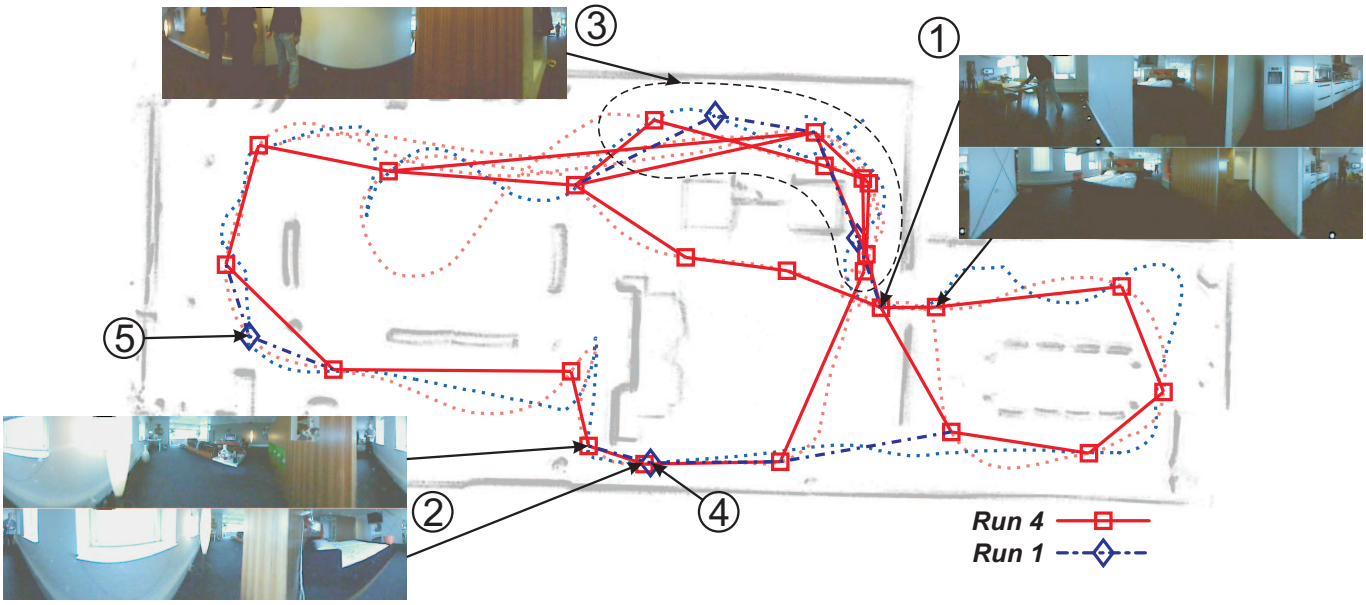

Fig. 2. Topological Map of the Environment created from the Datasets

B. Map Building

We test the performance of the map building algorithm using both datasets. We have concatenated *Run 1* onto the end of *Run 4* and processed both concurrently to test the performance of the proposed method at handling the kidnapped robot problem. The results obtained from the algorithm are shown in Figure 2. The locations of the topological map are superimposed on an approximate grid map generated from the accompanying laser range finder data. The dotted lines show the actual path of the robot while the solid lines form the connections between the various topological nodes. All the red information (squares) pertains to the *Run 4* dataset while the blue information (diamonds) pertains to the *Run 1* dataset. The algorithm handled the kidnapped robot problem flawlessly, immediately reassigning the location of the robot to the map node closest to the start of the *Run 1* dataset.

When initially processing the input images, the number of SIFT features extracted from each image had an average of 317.1 and a standard deviation of 103.3. After processing these features to create the topological map, the regions within the map had an average of 46.2 features with a standard deviation of 42.4. Considering that the statistical descriptors require twice the storage space of conventional SIFT features, the algorithm achieves a reduction of approximately 30% in the amount of data to be stored.

In Figure 2 we have highlighted several regions of interest. In regions 1 and 2 we have nearby map nodes as a result of large changes in the environment. These changes occur from the movement of the robot through "doorways". The associated sets of frames in Figure 2 show the changes in the environment from one node to the next.

In Region 3 many nodes were created in close proximity. This can be attributed to the lack of stable features in the environment as well as the presence of dynamic objects. The associated frame shows a rather featureless environment found within this region.

The two nodes marked 4 in Figure 2 are taken from roughly the same location but from different runs. The two frames from these nodes are shown in Figure 3. This figure highlights one of the problems of the SIFT feature extractor. The changes in illumination between the two runs have resulted in almost no matches between the two figures, despite them being from almost exactly the same view point. While the SIFT feature extractor claims to be photometrically robust, the change in illumination have caused saturation of the camera in some parts of the image, affecting the performance. SIFT feature matching performed using (1) yields two matches (one of which is incorrect) shown in yellow. Our method identifies 3 matches, shown in red, that match similar areas in the two images. Unfortunately, these matches were not sufficient to enable the localisation of the robot at point 4 within the map, resulting in two nodes being created at this point.

At point 5 in Figure 2 a new node was created on the

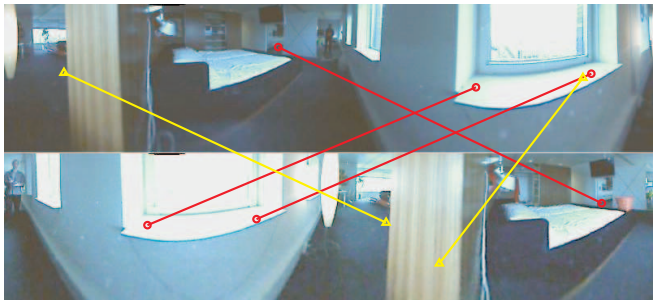

Fig. 3. Two images from different datasets at the same location showing the sensitivity of SIFT to illumination. Yellow matches △ are using the conventional SIFT matching method, while red matches ◯ are with our algorithm.

second pass. In this area, during the first pass in *Run 4*, there was an object in the environment that allowed a number of features to persist for much longer allowing a greater distance to be covered before the creation of a new node. On the second pass in *Run 1*, this object was removed from the environment causing a new node to be added sooner. This highlights the ability of the topological mapping algorithm to adapt to changes in the environment.

As shown in Region 3 of Figure 2, a lack of features in an environment presents the only significant problem of our method. We are currently investigating other means of feature extraction that may be used when an insufficient number of SIFT features are detected.

V. Conclusion and Future Work

We have introduced a vision based topological mapping and localisation algorithm that exploits the persistence of SIFT features within several omnidirectional camera images to improve data association. This algorithm also allocates topological regions conservatively, based on the number of visible features, as opposed to other algorithms that space regions equally in time or distance. This region definition coupled with the selection of only persistent features, from the myriad of features usually identified within an image, limits the total amount of information required to map an environment by 70% relative to other methods storing SIFT descriptors directly. We have also shown that the statistical approach to landmark matching can yield a better result than normally obtained using only single descriptors.

A. Future Work

At present we are working on a real-time implementation of this algorithm on a mobile platform with integrated navigation between regions. To improve the localisation performance, we are looking at using a statistical algorithm to predict the certainty of being in a region given previously occupied regions and the stored odometry relations. The algorithm may also be adapted to improve the location estimates of a map node using a SLAM infrastructure as described in [18]. Finally navigation is being implemented using the A* graph planning algorithm with the relative region displacements allocated during the mapping algorithm as edge weights to enable improved path planning.

Acknowledgements

The dataset used in this paper was obtained from the Robotics Data Set Repository (Radish) [17]. We would like to thank the providers of the Cogniron dataset for making their data available.

References

[1] T. Goedeme and M. Nuttin et al. Omnidirectional vision based topological navigation. *International Journal of Computer Vision*, 74(3):219–236, 2007.

[2] A. Tapus and R. Siegwart. Incremental robot mapping with fingerprints of places. In *Proc. IEEE Int'l Conf. on Intelligent Robots and Systems (IROS)*, 2005.

[3] A. I. Eliazar and R. Parr. DP-SLAM 2.0. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, pages 1314–1320, 2004.

[4] R. Sim and J. J. Little. Autonomous vision-based exploration and mapping using hybrid maps and Rao-Blackwellised particle filters. In *Proc. IEEE Int'l Conf. on Intelligent Robots and Systems (IROS)*, pages 2082–2089, 2006.

[5] N. Tomatis, I. Nourbakhsh, and R. Siegwart. Hybrid simultaneous localization and map building: A natural integration of topological and metric. *Robotics and Autonomous Systems*, 44:3–14, 2003.

[6] A. C. Victorino and P.Rives. SLAM with consistent mapping in an hybrid model. In *Proc. IEEE Int'l Conf. on Intelligent Robots and Systems (IROS)*, pages 3575–3580, 2006.

[7] J. A. Castellanos, J. Neira, and J. D. Tardos. Limits to the consistency of EKF-based SLAM. In *IFAC Symp. on Intelligent Autonomous Vehicles*, 2004.

[8] B. Kuipers. Modeling spatial knowledge. *Cognitive Science 2*, pages 129–153, 1978.

[9] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, pages 1023–1029, 2000.

[10] P. E. Rybski, F. Zacharias, and J. F. Lett. Using visual features to build topological maps of indoor environments. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, pages 850–855, 2003.

[11] O. Booij and B. Terwijn et al. Navigation using an appearance based topological map. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, pages 3927–3932, 2007.

[12] F. Fraundorfer, C. Engels, and D. Nister. Topological mapping, localization and navigation using image collections. In *Proc. IEEE Int'l Conf. on Intelligent Robots and Systems (IROS)*, pages 3872–3877, 2007.

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[14] E. Motard and B. Raducanu et al. Incremental on-line topological map learning for a visual homing application. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, pages 2049–2054, 2007.

[15] A. C. Murillo, J. J. Guerrero, and C. Sagues. SURF features for efficient robot localization with omnidirectional images. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, pages 3901–3907, 2007.

[16] A. Gil and O. Reinoso et al. Improving data association in vision-based slam. In *Proc. IEEE Int'l Conf. on Intelligent Robots and Systems (IROS)*, pages 2076–2081, 2006.

[17] Andrew Howard and Nicholas Roy. The robotics data set repository (radish), 2008.

[18] H. Andreasson, T. Duckett, and A. Lilienthal. Mini-slam: Minimalistic visual SLAM in large-scale environments based on a new interpretation of image similarity. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, pages 4096–4101, 2007.