Improving topological maps for safer and robust navigation

A. C. Murillo, P. Abad, J. J. Guerrero, C. Sagüés

Abstract— **Nowadays we frequently find big amounts of data to work with, what facilitates many robotic tasks and helps to solve perception problems. At the same time, this fact origins an interesting ongoing research problem: how to organize and arrange big sets of information to be useful in later uses. Topological mapping is a very useful tool to arrange and deal with big amounts of reference images for robotic tasks. There are many previous works on topological mapping and many others use this kind of maps for topological localization, planning and navigation. This work is focused on the problem of carefully design topological map building processes that facilitate the posterior robot tasks that use them and make them safer. We propose a new hierarchy of topological maps focused on this aspect. The experiments included in this paper were run outdoors using omnidirectional images and GPS information, and show the good topological maps obtained and how they allow robust and safer localization and navigation tasks.**

I. INTRODUCTION

Current autonomous systems are able to acquire large and detailed datasets of their environment, which allows them to obtain better interpretations and models of this environment. These issues also provide the robots with larger autonomy and capability of performing higher level tasks. Unfortunately, big amounts of data have also disadvantages: harder and more expensive computations are required to sort and make use of them. This problem is particularly important working with big image datasets, since they need powerful and intelligent designs to process them in a useful way.

In most robotic tasks, a basic step is to obtain a representation of the environment, by interpreting the sensory data acquired online or in exploration phases. Focusing on vision sensors, this task consists of arranging the acquired images into a visual memory or reference map. We need to organize the acquired data efficiently but more importantly, in a way as useful as possible to be used later. Big and accurate metric maps are often not necessary, so higher abstraction levels, e.g. topological or object-based maps are a good solution, at least on the top of a hierarchy of maps. This idea has been considered in hierarchical localization methods with a topological level on top of a metric one [1], [2].

Our work is focused on how to improve a hierarchy of topological maps: how do we build a topological map we can use later in the most efficient, robust and safe way possible? This paper presents our proposal to improve typical topological mapping techniques with a series of steps focused on the later usage of that map.

Some previous works try to integrate the topological map building with its posterior use, such as the works in [3] or [4], where they demonstrate how to navigate using the maps they build. In [4], their authors also pointed that the performance of tasks such as localization and navigation, depends tightly on the way the reference maps are built. Then, we should pay special attention to this building process: e.g, if the topological map is composed by very distant reference nodes (images), localization on this map would be very hard, even impossible, without a very powerful wide baseline matching technique.

Typically we find two types of topological mapping approaches, offline or online, with clear advantages and disadvantages for each type, commented in more detailed in next section. Our proposal runs offline, since we need to have data from the whole environment to indicate what we consider safe areas to drive. However, we try to include interesting properties typical from online approaches, taking into account the temporal consistency of the images, since they were acquired sequentially. Another idea proposed is to augment the typical map levels used in hierarchical approaches, metric and topological, by sub-dividing the topological level into two different ones: a coarser one for topological localization and a more detailed one for safer navigation.

Fig. 1. Robot and sensors used, omnidirectional camera and GPS receiver.

The experiments demonstrating our approach use a robot equipped with an omnidirectional camera and a self-positioning system (see Fig. 1), in this case GPS, but others such as an improved odometry indoors would work as well. Although we use GPS information to build the map, it can be used as reference by other robots without GPS receiver.

Section II presents a brief comparison of previous topological mapping works, section III describes our proposal and section IV shows the results obtained with it. Finally section V concludes the work.

II. TOPOLOGICAL MAP BUILDING

Topological map building is an old subject of interest in the robotics field. Many interesting results have been

A. C. Murillo, P. Abad, J. J. Guerrero and C. Sagüés are with the I3A/DIIS at the University of Zaragoza, 50010 Spain. acm@unizar.es

This work was supported by projects DPI2006-07928, DPI2009-14664.

presented for a long time, not only based on vision sensors [5], but also using for instance range sensors [6]. Initially, topological maps were a particularly useful tool to facilitate planning and navigation. However, in recent contributions regarding topological mapping, we find additional motivations.

On one hand, the need of tools to deal with big datasets, in particular image sets, and efficiently process and use them. Indeed, grouping reference images into topological nodes or clusters and keeping only the corresponding centroids allow a more efficient use of the reference information: e.g., in visual based localization, there is no need to compare a certain query view with all the reference set but only with the cluster centroids. In this regard, recent hierarchical approaches for topological map building have shown nice results [7] [8]. Numerous recent works in the field of computer vision have also presented interesting results on how to classify, arrange and represent big sets of images [9], [10]. They aim the same goal, to facilitate the use of reference data in posterior tasks, such as visual localization or location recognition.

On the other hand, we tend to provide autonomous devices with higher abstraction concepts of their environment. Topological and cognitive maps are a good approach towards it [11] [12], providing easier interaction with humans and augmenting the kind of concepts and decisions the robots can deal with.

There are two big groups of topological mapping approaches: offline and online approaches. First one tries to optimize the image clustering once the whole data set has been acquired [7]. This presents the advantage of being able to get an overall optimum, but it has the disadvantage of being offline and usually computationally expensive. Online approaches instantiate a new cluster set each time the algorithm detects a significant change in the image acquired. Many different criteria has been studied to define what a significant change is: sometimes the partition is just of small subsets along the image sequence, while other times there is a complex decision process. These are usually less accurate but more efficient and they allow the map to be obtained as the robot moves as proposed in [13], [14].

In the approach proposed in this paper, we try to make use of the advantages of both types of methods. First we apply an offline approach to get a good estimation of the overall clustering once the data set is acquired, based only on the appearance of the images. Afterwards, we apply a filter that takes into account that the images where taken sequentially, and therefore consecutive images have higher probability of belonging to the same cluster. This is an issue that most online methods take into account implicitly. More details on our proposal are given in the following section.

III. Enhancing topological maps for safer navigation

As mentioned previously, our proposal aims improving the way of building topological maps, in order to facilitate mobile robots to use them later. Typically the topological map is integrated as one level of a hierarchical localization systems, divided into metric and topological steps. Here we propose to sub-divide this level into two other levels of accuracy, obtained as detailed in the following points.

A. First level of the topological map: clustering.

This first step aims an image clustering based on image similarity, with some interesting characteristics in the process, such as automatic selection of number of clusters and an *online filter* to take into account the temporal continuity of the reference images since they came from a sequence.

Once the reference images have been acquired in a guided exploration tour, they are organized following the next steps.

1) Local features and correspondences: SURF [16] features are extracted for all images. Then, correspondences between each pair of images are established using a standard approximate nearest neighbour technique together with a fast robustness filter to check consistent rotation between all feature correspondences.

2) Image similarity evaluation: similarity between pairs of images is obtained using the following expression.

$$DIS = \frac{1}{2}\frac{m \cdot d}{len}(\frac{1}{f1} + \frac{1}{f2}) + P_0\frac{(f1-m)}{f1} + P_0\frac{(f2-m)}{f2} \quad (1)$$

DIS is a dissimilarity measure, where m is the number of matches, fi the number of features in the image i, d is the average Euclidean distance between the descriptors of the matched features, len the length of the descriptor vector and P_0 a penalization weight to the amount of non-matched features. P_0 has been experimentally set to a value of 1.6. To transform DIS into a similarity measure normalized in $[0, 1]$, we define the following:

$$SIM = e^{\frac{-DIS}{(\max(DIS))}}. \quad (2)$$

3) Initial clustering: we use the graph cuts based clustering technique from [7] with the self-tuning approach from [15], that allows automatic detection of the best number of clusters for the dataset. This methods require to turn into binary the similarity measure values. Then, a threshold, typically from 0.3 to 0.4, is established for our similarity measure (2).

4) Online-filter: clustering results are refined with the proposed *online filter*, what prunes possible clustering misclassifications in a simple way. It follows the temporal order the images have been acquired, they all come from the same sequence, and if an image is clustered in a different node than previous and next images, it is considered a likely mistake and therefore changed to same cluster as the neighbours.

5) Representative image selection: finally, a few images to represent each cluster are selected, these are the topological map nodes. We select for each cluster its centroid image and the furthest image from this centroid. Keeping two images per cluster allows more robust visual topological localization afterwards. This choice also facilitates a final metric localization step using some structure from motion technique between multiple views, such as the metric localization run in the hierarchical localization from [2], where robust correspondences between current view and reference images are needed.

A few examples of the grouping obtained after these clustering steps are shown in Fig. 2.

Fig. 2. Two of the clusters obtained for the datasets used in the experiments.

B. Second level of the topological map: the navigation map.

Second issue dealt with this approach is to determine the connectivity among cluster representative images, cluster centroids for short, and therefore the possibility of navigation between them. The goal is to establish possible trajectory paths for the robot between the different reference locations explored (the different clusters). We need to decide where and how many way-points are necessary, besides cluster centroids, to have a safe navigation graph. To obtain this navigation map, we take into account not only the visual information but also the positioning information from the GPS tags associated to the images.

1) Clusters similarity evaluation: first, a similarity evaluation between clusters is performed, according to the number of local feature correspondences between the images that compose each of them, with a bonus if their GPS locations are close to each other. More formally written, if C_a and C_b are the set of images from clusters a and b respectively, the visual similarity among those clusters $V(a,b)$ is defined as

$$V(a,b) = \frac{\sum_{i \in C_a} \sum_{j \in C_b} S_{ij}}{|C_a||C_b|}, \tag{3}$$

with $|C_a|$ and $|C_b|$ the number of elements in C_a and C_b and S_{ij} the elements of the similarity matrix built previously to perform the initial clustering, see eq. (2).

We first obtain the appearance similarity between clusters and then add a bonus B to it depending on the Euclidean distance, $D(a,b)$, between centroid GPS locations of clusters a and b:

$$\textbf{if} \quad D(a,b) < kD_{img}(|C_a| + |C_b|) \quad \textbf{then} \quad V(a,b) = V(a,b) + B,$$

with k a value between 0 and 1 depending on how strong we want this filter to be, and D_{img} the average distance between every two consecutive images in our sequence.

2) Establishing connections: every couple of centroids that have a similarity $V(a,b)$ over the established threshold are considered connected, and this path is added as a new arc in the navigation graph.

To improve these initially established connections, two simple additional filters are defined based on the GPS measurements. The first one analyzes the distance between the two most similar images we can find taking one from each cluster of a certain connection. If this distance is too big, we break the connection between these clusters.

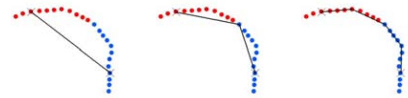


Fig. 3. Small example of the Split&Merge filter, to improve how a certain line segment fits to a trajectory.

The second filter checks that the following condition is true for every pair of connected clusters:

$$D(a,b) < D_{img} \cdot (|C_a| + |C_b|). \tag{4}$$

In simple words, it checks that the distance between two cluster centroids is below the maximum possible theoretical distance: according to the average separation between two consecutive images in the original sequence D_{img} multiplied by the total number of images in both clusters $|C_a| + |C_b|$.

3) Establishing safe way-points through a Split&Merge algorithm: finally a process is run to adjust the navigation graph arcs, allowed trajectories, to previously visited paths. Here the arcs are fitted as close as possible to the trajectories followed in supervised exploration since we know those are *safe* terrain. This way we can avoid our robot to navigate into dangerous surfaces, not easy to detect with typical reactive navigation systems, such as water, dense grass or small stones areas. To achieve this, we apply the well know Split&Merge algorithm for line fitting [17] to the set of points composed by the GPS locations of the reference images used to build the map. Fig. 3 shows a brief example on how this algorithm improves an initial line set to fit better to a particular point set, establishing as many extra segments as necessary. More details on results obtained with this filter can be seen in the experimental section.

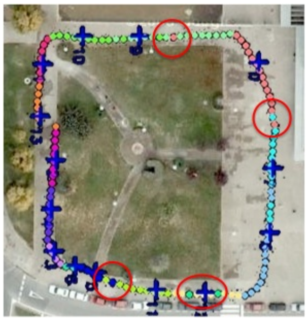
IV. Experimental Results

This section presents several experiments to test our proposal in a real campus outdoor environment. All experiments were performed around the same area to facilitate the localization and navigation tests with our robots on the same areas the topological maps were built. Three different datasets were acquired in this environment. All of them consist of a sequence of omnidirectional images, acquired with a conventional camera pointing to an hyperbolic mirror (see Fig. 1), and the GPS tag associated to each image, acquired with a differential GPS sensor. Note the GPS signal is not always very accurate when navigating close to buildings but most of the time it helps a lot for building an improved map. The datasets used are:

CPS1: 130 images acquired during a 200 m. loop.

CPS2: 135 images from an open trajectory of around 240 m. Only every 5^{th} image is used to build the map.

CPS3: 306 images acquired during a more complex trajectory of around 500 m, with several loops included.



1-Topological map after only image clustering step.

2-Topological map after *online-filter* is applied.

3-Topological and Navigation map based only on cluster centroids.

4-Topological and Navigation map after *Split&Merge-filter* is applied.

Fig. 4. Test CPS1: topological and navigation maps obtained at different steps of the method. Different colors represent each of the clusters that to compose the topological map. White connections between reference positions are the navigation map graph arcs. Blue crosses with numbers by their side are initial cluster centroids, while crosses without numbers are new way-points established for safety. On the left images, red circles point misclassifications or dangerous paths that are automatically fixed along the process. (Best viewed in color)

A. Map Building

These experiments show the improvements in the topological map built following our proposal with regard to the results obtained with the image clustering techniques used as basis. The following results are a summary of the topological map building tests run with the datasets mentioned before. All experiments were run with the same topological mapping process explained previously, using as local features SURF [16] with 64-length descriptor.

1) Test CPS1: Fig. 4 shows the evolution of the topological and navigation maps obtained at the different steps of the proposal. We can observe how the *online filter* helps pruning misclassifications from the offline image clustering step (top images in the figure), and how the *Split&Merge* based step improves the coverage, robustness and safety of the lower level navigation map (bottom images in the same figure).

2) Test CPS2: Fig. 5 shows the final results of the topological map obtained in this second test. We can observe a clean final clustering for the top level graph (topological map) and a clean navigation map that covers all the explored area without dangerous transitions, similarly to the previous test but in a slightly bigger environment.

3) Test CPS3: this final and more complex test demonstrates how the approach still works fine with complex trajectories. Fig. 6 presents the topological and navigation maps obtained at the beginning and at the end of the process. We can observe that many misclassified areas are corrected and many dangerous connections established in the initial navigation map are avoided at the end of the process. Besides, we should note that the process successfully detects some revisited areas, see for example the areas marked with a green circle on the middle of Fig. 6.

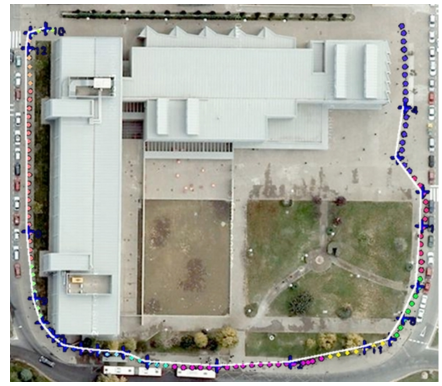


Fig. 5. Test CPS2: final topological map (each cluster represented by a color) and navigation map (white connections between reference locations) obtained after applying all steps of the proposal. (Best viewed in color)

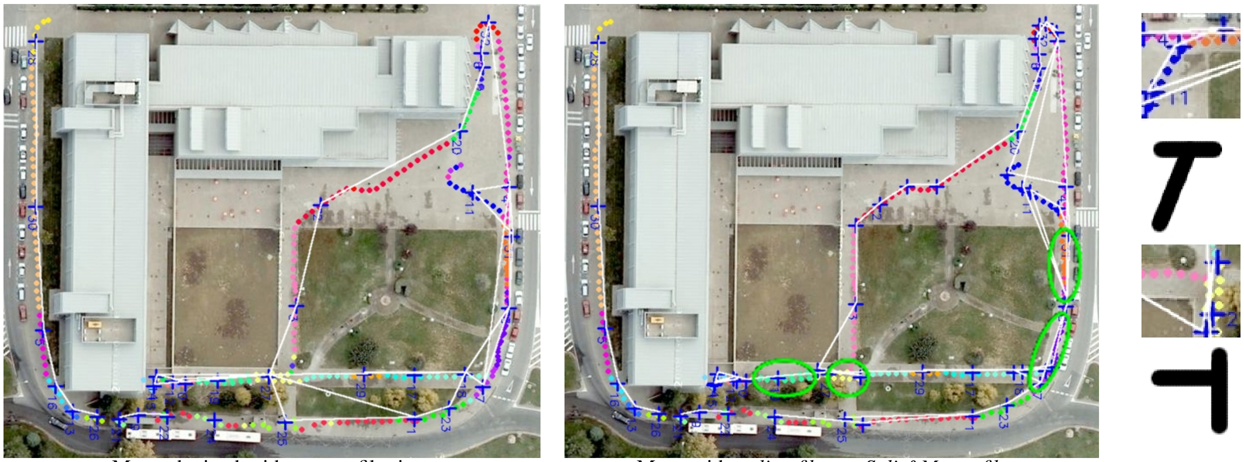
This last test is also useful to show examples of two problems that can occur using this approach and that point future work directions.

First, there are particular trajectory configurations, T-junctions, where line fitting with the Split&Merge approach does not work properly. Two examples of T-junctions in our trajectories are shown on the right details of Fig.6. At those corners obtained arcs in the navigation graph are not as good as for the rest.

Secondly, we can observe an example of how bad GPS measurements can spoil the image clustering, because GPS distance is used as a weight to try to join neighboring images. Then, if at some point that signal is bad the *offline filter* does not work as well as in the rest of the areas. Left of Fig. 7 shows the quality of the GPS signal used in this experiment. Note the right bottom corner area, not only the strength of signal is not good, but big jumps can be observed in the measurements, while actually the robot was just driving straight. On the middle of the same figure, we see a not very clean clustering in this area, and several wrong navigation connections. Right of Fig. 7 shows how the same area, but with data from a different acquisition where the GPS did not fail, can be properly represented with a correct topological and navigation map.

B. Map Usage

This second set of experiments intends to show how the topological map building proposal actually helps in posterior tasks of localization and navigation, improving their robustness and safety. The top level of our map hierarchy, the topological map, is essential for an efficient localization. The lower level navigation map is essential for a safe and autonomous navigation, either using vision or range sensors.



Maps obtained without any filtering Maps with *online-filter* + *Split&Merge-filter*

Fig. 6. CPS3-dataset: Topological and Navigation maps obtained at different stages of the method for the more complex dataset. On the right, details of T-junctions where the navigation map construction fails to leave 100% safe navigation way-points.

Fig. 7. Left: quality of signal at the GPS receiver. The quality increases from red (bad) - yellow - blue - green (best). Middle-Right: clustering results obtained in an area with bad GPS signal, and results obtained in the same area using other acquisition where GPS coverage was better.

The following experiments have been run with dataset CPS3, since it is the more complex one.

1) Localization: GPS receiver is used in the exploration stage to build a better and more useful map. However, not all our robot team members, that are going to localize themselves in this map, can always be equipped with good GPS sensors. For the localization tests, a set of test images, different from those used to build the map, are compared to the cluster centroids of the topological map. This similarity evaluation has been done following the approach previously presented in [2], based on global and local image features.

Since we have GPS tags with common reference frame, we can plot a nice summary of all the localization results as shown in Fig. 8, where we can get an overall idea of the obtained localization results. Red * represent the location of the evaluated test images, and blue $<$ represent the location of the centroids of the topological map clusters. Blue lines join every test with its selected cluster centroid, most of them are correct (98% correct localization results). There are only two mistakes (marked with a red line), and a strange case that corresponds to a correct localization (green line): the approach evaluates properly the similarity but the query had a very noisy GPS tag, so it seems to be very distant in the plot. We must say that the images were taken under similar weather conditions, what helps in the feature correspondence search. The localization ratio would probably decrease if test images were taken under very different conditions than the ones used to build the topological map.

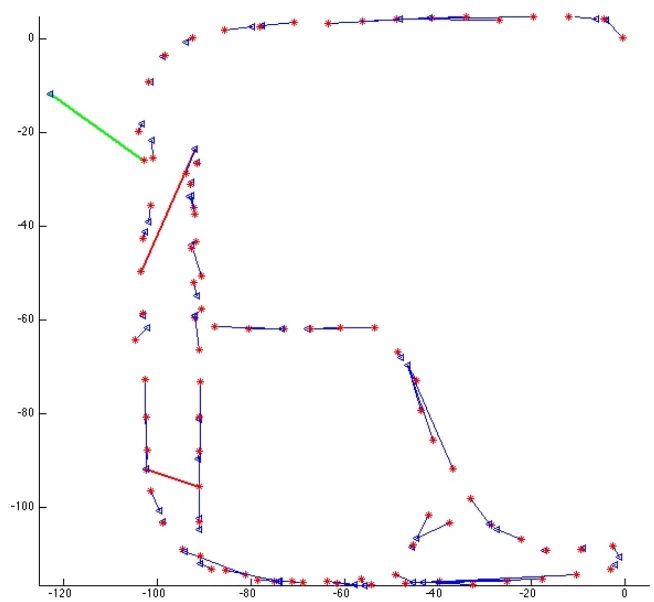


Fig. 8. Localization results. Location of query images in Red *. A line connects each query with the cluster centroid (blue $>$) where the localization estimates that it is located. (Best viewed in color)

2) Navigation: the planning to go from one place to the goal location is done with the Dijkstra algorithm, to find the fastest path according to the approximate distance between the nodes that compose the navigation graph. For navigation, it is very important to predict in advance dangerous situations for the robot, such as driving it into dense grass or stone

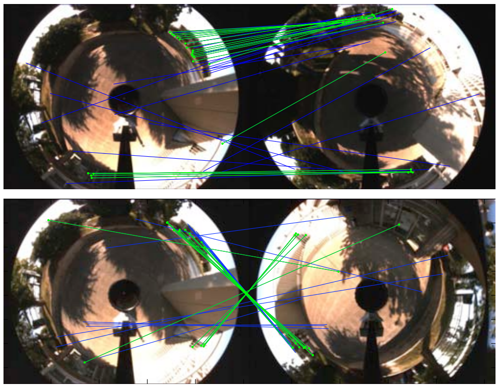


Fig. 9. Example of robust correspondences between one navigation node (left) and two other nodes connected to it in the navigation graph. Blue lines: all tentative matches. Green lines: robust matches obtained after estimating the epipolar geometry between the two views (best viewed in color).

areas, or even worse such as water. Besides checking this safety issues, we have evaluated how useful this map would be for two types of navigation.

Navigation based on range sensor. We have successfully used the navigation map with a reactive navigation approach, ND-navigation [18], based on a range sensor. It allows the robot to move from current location to the goal location, one of the navigation map way-points. We just need to provide the GPS tags of the nodes from the navigation graph that we need to traverse until the goal location. The only issue here is to make sure that all way-points can be reached safely, since the ND-algorithm automatically takes care of static and dynamic obstacles.

Vision based navigation. The essential issue to perform vision based navigation is to obtain enough robust feature correspondences between every two way-point images we need to go through. We have made some successful tests to extract enough robust correspondences between the navigation map way-points connected in the navigation graph. Fig. 9 shows an example of robust SURF correspondences between omnidirectional image pairs that correspond to connected nodes of the navigation map. They are obtained through the estimation of the epipolar geometry between both images. If we are able to obtain relatively big sets of robust correspondences between two connected nodes, it is feasible to navigate between them using a standard visual servoing approach, based on local feature correspondences and epipolar geometry constraints. To carefully test this part we intend to integrate the process with an omnidirectional image based servoing technique such as the one used in [4].

V. Conclusions

We have presented a new approach to improve the way of building topological maps, so that the posterior localization, planning and navigation tasks can be performed as efficiently and safely as possible for the robot. One of the ideas in the proposal is to decompose the topological map on two levels. The first one composed of image clusters, based mainly on appearance similarities, which is very important for a correct posterior localization on this map. The second level, a navigation map or graph, that analyzes the exploration path followed by the robot to establish safe transitions and additional way-points if necessary to cover all the locations included in the topological map. As future work, we aim to integrate all the described steps with a visual servoing system to test if the visual based navigation can be as safe as we already checked with range sensors based navigation. We also intend to study deeply the ideas of a hierarchy of topological maps at different semantic levels, each of them built paying attention on some specific issue.

Acknowledgements

The authors would like to thank L. Riazuelo for his help in the experiments, D. Viu for his initial tests and implementations and the reviewers for their suggestions.

References

[1] 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.

[2] A. C. Murillo, C. Sagüés, J. J. Guerrero, T. Goedemé, T. Tuytelaars, and L. Van Gool. From omnidirectional images to hierarchical localization. *Robotics and Autonomous Systems*, 55(5):372–382, 2007.

[3] O. Booij, B. Terwijn, Z. Zivkovic, and B. Kröse. Navigation using an appearance based topological map. In *Int. Conf. on Robotics and Automation*, 2007.

[4] T. Goedemé, M. Nuttin, T. Tuytelaars, and L. Van Gool. Omnidirectional vision based topological navigation. *Int. J. Comput. Vision*, 74(3):219–236, 2007.

[5] J. Santos-Victor, R. Vassallo, and H. Schneebeli. Topological maps for visual navigation. In *Int. Conf. on Computer Vision Systems*, pages 21–36. Springer-Verlag, 1999.

[6] S. Thrun and A. Bucken. Integrating grid-based and topological maps for mobile robot navigation. In *AAAI/IAAI, Vol. 2*, pages 944–950, 1996.

[7] Z. Zivkovic, B. Bakker, and B. Kröse. Hierarchical map building and planning based on graph partitioning. In *IEEE Int. Conf. on Robotics and Automation*, pages 803–809, 2006.

[8] A. Štimec, M. Jogan, and A. Leonardis. Unsupervised learning of a hierarchy of topological maps using omnidirectional images. *Int. J. of Pattern Recognition and Artificial Intelligence*, 22(4):639–665, June 2008.

[9] C. Siagian and L. Itti. Storing and recalling information for vision localization. In *IEEE Int. Conf. on Robotics and Automation*, 2008.

[10] K. Ni, A. Kannan, A. Criminisi, and J. M. Winn. Epitomic location recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[11] A. Tapus and R. Siegwart. Incremental robot mapping with fingerprints of places. *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 2429–2434, 2005.

[12] S. Vasuvedan, S. Gachter, V. Nguyen, and R. Siegwart. Cognitive maps for mobile robots - an object based approach. *Robotics and Autonomous Systems*, 55(5), 2007.

[13] C. Valgren, T. Duckett, and A. J. Lilienthal. Incremental spectral clustering and its application to topological mapping. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 4283–4288, 2007.

[14] Xuming He, Richard Zemel, and Volodymyr Mnih. Topological map learning from outdoor image sequences. *Journal of Field Robotics*, 23(11-12):1091–1104, 2007.

[15] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in neural information processing systems*, 2004.

[16] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *The ninth European Conference on Computer Vision*, 2006, http://www.vision.ee.ethz.ch/ surf/.

[17] G. de Araújo Borges and M.J. Aldon. A split-and-merge segmentation algorithm for line extraction in 2-d range image. In *Int. Conf. on Pattern Recognition*, pages 1441–1444, 2000.

[18] J. Minguez and L. Montano. Nearness diagram navigation (nd): Collision avoidance in troublesome scenarios. *IEEE Transactions on Robotics and Automation*, 20(1):45–59, 2004.