Navigation using an appearance based topological map

O. Booij, B. Terwijn, Z. Zivkovic, B. Kröse
Intelligent Systems Lab Amsterdam
University of Amterdam
1098 SM Amsterdam, The Netherlands
Email: obooij@science.uva.nl

***Abstract*— Vision systems are used more and more in 'personal' robots interacting with humans, since semantic information about objects and places can be derived from the rich sensory information. Visual information is also used for building appearance based topological maps, which can be used for localization. In this paper we describe a system capable of using this appearance based topological map for navigation. The system is made robust by using the epipolar geometry and a planar floor constraint in computing the necessary heading information. Using this method the robot is able to drive robustly in a large environment. We tested the method on real data under varying environment conditions and compared performance with a human-controlled robot.**

I. Introduction

Recent developments in the field of 'personal' robots, which interact with humans in a natural way, bring new insights in the representations needed by the robot to fulfill its task.

For example the internal model of the environment, used by the future home robot for goal-directed navigation, must contain spatial concepts understandable for the human. In an indoor environment typical classes such as 'rooms', 'objects' or 'doors' must be detected.

The sensing system of the robot must be able to distinguish between different instances of these classes. The traditional sensors on robots such as laser range finding — mainly used for obtaining geometric information and navigation — have been used for semantic labeling of places [1], but a more appealing solution is to use vision. Abstract visual cues have been used in classifying rooms [2], [3] and recently representations based on visual object recognition have been presented [4] for spatial descriptions.

Apart from semantic labeling, visual information can also be used for map building. In [5] a 3D representation is built, where locations of distinctive features are reconstructed. Recently we have presented an approach where the environment map is not a 3D reconstruction but is represented as an 'appearance graph': a topological representation where nodes represent omnidirectional images taken by the robot and edges are defined by similarities between these images [6]. We showed that this representation can be used for path planning [7] and for finding a categorical representation [8]. In this paper we will show that this representation can be used for navigation.

Strategies for mobile robot navigation with omnidirectional vision systems have been reported earlier. [9] shows the epipolar constraint can be used for moving from one pose to another in a simulator. Navigation over longer paths has been reported by [10], but this system is restricted to travel only along prerecorded trajectories.

In this paper we present visual navigation using the appearance graph, making it possible to drive trajectories not driven in the training phase. In Section II we will first summarize our appearance based topological representation. Then we will explain our navigation strategy, in which we use the epipolar constraint and the constraint of moving over a planar ground floor to obtain a robust heading estimation (Section III). In Section IV we explain how to navigate over the graph representation. Experiments on orientation estimation and navigation using real data are reported in Section V.

II. Appearance based topological mapping

In this section we describe the method used for constructing the appearance based map. This has already been reported in [6]. The goal is to construct a weighted graph $G = (V, S)$ in which the nodes V denote images taken at certain positions in the environment and each link S_{ij} in the graph denotes that image i and j look similar and are thus likely to be taken from more or less the same position [11]. The similarity measure we use is directly linked to the ability to perform navigation between the two positions. If we can robustly reconstruct the local geometry given the two images then we define a link $S_{ij} > 0$ between the two nodes. The robustness of the reconstruction is expressed in the value of S_{ij} of the link.

We start with a set of images taken by the robot while it was driven around in the environment. In order to get a large overlap in the images the robot is equipped with an omnidirectional vision system consisting of a hyperbolic mirror and an ordinary camera (see [12] for details). From each of the resulting panoramic images a set of SIFT features is extracted [13]. Then for each pair of feature sets, corresponding points are found by comparing their SIFT descriptors, see Figure 1. The epipolar geometry is determined using robust estimation techniques, which are also used for robot navigation (more about this in Section III). One of the outputs of the estimation method is the number of point correspondences that agree with the epipolar constraint. However from these correspondences there is still a percentage of false feature matches. The number of these false matches is in the order of the

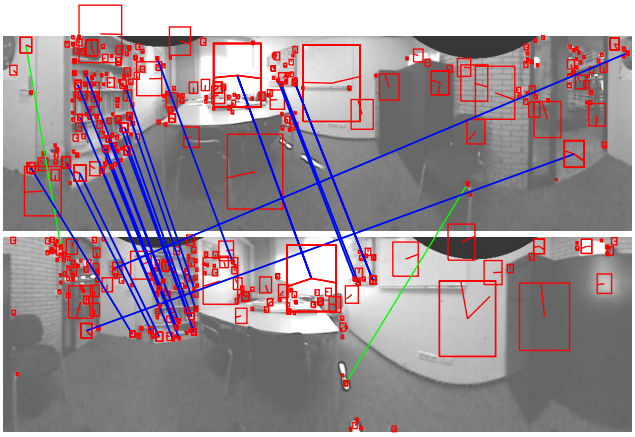

Fig. 1. Matching two images. The red boxes indicate the SIFT features found in the images. The lines connecting two of these features indicate that they correspond. If the line is blue, this means the corresponding pair agrees with the epipolar constraint. If it is green, it does not agree with the epipolar constraint and is thus probably an outlier.

number of features found in the two images. By dividing the number of constrained point correspondences by the lowest number of features found in the two images we obtain a similarity measure between 0 and 1. If this value is larger than a certain threshold, which indicates the robustness of the local geometry estimation, the images match and a link S_{ij} is created between the two nodes representing the images with its value set to the similarity.

In Figure 5 an appearance based map is shown as constructed for the navigation experiments. The map is purely topological, as there is no explicit distance or scale information present in the graph (note that the position of the nodes is based on odometry information, however this is only used for visualizing the graph.) What is contained in the map is information on the neighborhood relation of different parts of the environment. The graph representation is well suited for further processing. In [6] we explain how graph clustering techniques are used to find convex spaces in the map, which correspond to rooms and corridors in the environment. Augmenting the graph-clusters results in a semantically labeled map, which can be used for human robot interaction [14].

III. Heading estimation using the epipolar geometry

Epipolar geometry estimation is thoroughly discussed in computer vision literature and some standard implementations are readily available [15], [16]. However, because we use an omnidirectional vision system and a robot to obtain the images, there are some special issues to take into account.

It is taken that we have extracted a set of N matching point pairs from the two panoramic images. The image points are then projected on a sphere around the optical centers with distance 1. Let us denote the 3D points in the current image as $\{\mathbf{x}_1^{(1)}, ..., \mathbf{x}_1^{(N)}\}$, and the corresponding points in the target image as $\{\mathbf{x}_2^{(1)}, ..., \mathbf{x}_2^{(N)}\}$. Omnidirectional vision systems

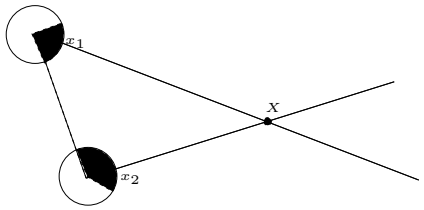

Fig. 2. A 2D visualization of a corresponding point in two panoramic images. The circles denote the panoramic images and the black dot is the worldpoint X. The solid lines are the oriented rays going through the optical center and the worldpoint. The filled parts of the circles denote those places in the images where the oriented ray of the other camera can be projected, i.e. the epipolar line. As can be seen in only a small part, less than 50% of the image, a corresponding point can be found.

are by default calibrated in order to produce single viewpoint images. So we can use the essential matrix E, instead of the more general fundamental matrix for uncalibrated images, to relate point correspondences in the following way:

$$(\mathbf{x}_1^{(i)})^T E \mathbf{x}_2^{(i)} = 0 \qquad \text{for all } i. \tag{1}$$

Using 8 point pairs we can linearly solve E with the 8-point algorithm [15]. However the robot is driving over the planar ground floor. Hence, it can be assumed that the positions of the images do not differ in height and the relative rotation only occurs around the vertical axes. This prior knowledge can be incorporated by restricting the essential matrix in the following form [17], [18]:

$$E = \begin{bmatrix} 0 & e(2) & 0 \\ e(4) & 0 & e(6) \\ 0 & e(8) & 0 \end{bmatrix} \tag{2}$$

The minimal number of point pairs for the linear estimation of this constrained essential matrix is only 4 and because the solution space is smaller the estimator is less effected by noise.

The essential matrix bears the relative rotation R and translation $\vec{t}$ up to an unknown scale between the positions of the two images as follows:

$$E = RS, \tag{3}$$

where S is a skew-symmetric matrix composed of the elements of $\vec{t}$. The essential matrix can be decomposed into 4 different solutions of $\vec{t}$ and R. By imposing the constraint that world points should lie in front of the image surface on which it is projected, we can choose the correct solution [19]. The world points that were projected behind one of the image surfaces given the correct R and $\vec{t}$ were obviously produced by false matching. For panoramic images the chance that a false match is in front of both image surfaces is small, because omnidirectional vision systems look in every direction, see Figure 2. We use this knowledge in the robust estimation process described below.

Generally image points are not noisy-free and part of the point pairs found by the matching algorithm is the result of false matching. Therefore a simple least squares method to fit an essential matrix to the data will fail miserably.

A fast and robust estimation method that can cope with a large percentage of these false matches is RANSAC (random sample consensus), which we use to determine correct point pairs [15], [16]. RANSAC estimates a large number of essential matrices and chooses that E that agrees with the most point pairs. In each run it first estimates E given 4 randomly chosen point correspondences using the planar version of the 8-point algorithm. Then we check if the four correspondences all lie in front of both image planes. If not, then we discard that estimate of E (this type of model checking was first proposed in [20]). If the correspondences are consistent, we use E to reproject all point correspondences of the image pair and count the number inliers. An inlier in our case, is a correspondence that has a low reprojection error and lies in front of both cameras.

After choosing the run with the highest number of inliers, a final E is computed by taking into account all its inliers. From this E, the R and $\vec{t}$ (up to a scale factor) are determined between the image locations[19]. In the remainder of the paper we assume that, if R and $\vec{t}$ can be determined robustly, the robot can indeed move from one location to another. This need not be true, for example if there are obstacles with very few localized features, or if the features are located in a restricted region of the images. In our experimental setup we therefor had a local obstacle algorithm operational using sonar. In the runs we present in this paper we did not need the obstacle avoidance.

The heading ϕ the robot has to drive when navigating from the current image to the target image can be calculated using

$$\phi = \text{atan2}(t_y, t_x). \tag{4}$$

IV. Navigation over the topological map

In this section we describe the framework to navigate to a goal location in the environment mapped by the appearance based graph given the heading estimation explained in section III. The general aim is that the robot should be able to navigate to any room in a building by giving it a node in the graph. In our specific case we desire that the robot is able to match the last observation with that of the goal node.

A challenge is that there is no positional information stored in the representation. Thus, two images, whose nodes are neighbors in the graph, could have been shot at any distance from each other. Techniques exist to estimate the distance from two images. However these techniques would require us to make an assumption on the position of landmarks in the world, making the system less flexible. Another solution is to find corresponding features in three or more images, which is quite common in the field of visual servoing (see for example [21]). However, in dynamic environments it will be more difficult to find stable correspondences in three images than in just two.

We take it that the goal location is given by a node in the graph. First Dijkstra's shortest path algorithm [22] is used to compute the distance D_i from every node i in the graph to this goal node. This algorithm requires the links of the graph to be labeled with a distance measure while we have a similarity measure. Therefore we define the distance d as $d_{ij} = \frac{1}{S_{ij}}$. The distances of the nodes to goal node are used during driving as a heuristic to drive in the direction of the goal node.

Note that the algorithm in our case gives the shortest path in the appearance space, which is not necessarily the shortest path in the metric space. Because of our distance measure, a shortest path will favor a selection of image sequences which have many features in common. This may imply that the robot will avoid path elements where the local features change rapidly (close to narrow throughways) and prefer to navigate in the center of large open spaces. We plan to design experiments to test this in more detail: in this paper we focus on the robustness with respect to occlusions.

The navigation procedure directs the robot to one node at a time that can be seen as a subgoal node on a path to the goal node. This path of nodes could have been planned in advance. However this would result in a very inflexible trajectory which would be difficult to traverse in a dynamic environment. In the following we explain how the subgoal nodes are determined dynamically while driving.

At the start of the trajectory the robot localizes itself in the appearance based graph by taking a new observation and comparing it with all the images in the graph following the same matching procedure as used for constructing the graph (see Section II). The node of the graph with the highest similarity is chosen as the current subgoal node c of the robot. This procedure is linear in the number of nodes and could thus be time consuming.

If a subgoal node is determined the robot tries to pick a new subgoal by comparing the newest observation with all the neighbors of node c that have a smaller distance D_c to the goal node. If one of these images matches, it becomes the new current subgoal c. This procedure is repeated for the neighbors of the new c, until the node is found that is closest to the goal node and does still robustly match the new observation.

When a subgoal is determined, the heading is estimated in order to drive in its direction. This heading will not be perfectly directed toward the subgoal, partly because of sensor-noise, but also because the environment could have changed after the appearance based map was constructed. Therefore a recency weighted averaging filter is used which to takes into account previous estimates of ϕ.

This smoothed heading is now used to move the robot. It then takes a new observation while driving and repeats the whole procedure. This goes on until the subgoal is equal to the global goal and the robot is stopped, completing the navigation.

We also need some recovery method in case the robot gets lots. It could happen that the robot is repeatedly unable to estimate the heading with the current subgoal, because it finds less than 4 corresponding image points. This can be due to changing environmental conditions, but can also be caused by bad heading estimates for the previous observations. If the robot can not find a heading for 10 observations in a row it will try to relocalize itself in the map and start with the new

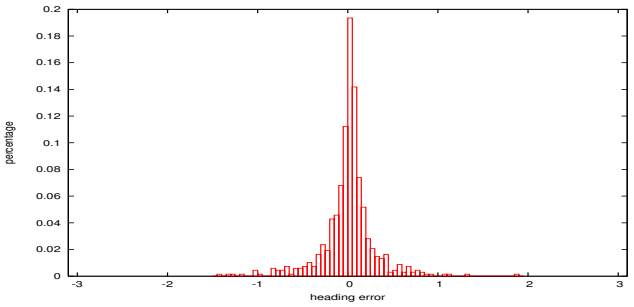

Fig. 3. Histogram of the differences between the estimated and the ground truth heading of all the pairs of images.

node as subgoal.

V. Experiments

For the experiments a Nomad Super Scout II is used which is equipped with an omnidirectional vision system consisting of a hyperbolic mirror and an ordinary camera. The navigation procedures are tested in an office environment.

We first test if the heading estimation works properly by comparing it with ground truth positioning data of the robot. Then a large appearance map is constructed by driving the robot manually through the environment. This map is used for the navigation experiments, in which we compare the length of the traversed path to that of a manually driven path. Also we test the robustness against noise, by obstructing part of the view of the robot while it is driving.

A. Heading estimation

First the low level heading estimation is tested by comparing it with the ground truth positions and orientations of the robot. A small data set is taken by the robot on a 3 by 3 grid of approximately 2 square meters in size. On each point of the grid 4 images are taken with the robot facing in 4 different directions, giving a total of 36 images.

For each pair of images the heading is computed given the method explained in sections II and III. The headings between images taken at the same location are meaningless and thus ignored. The estimated heading is compared with the ground truth heading calculated on the basis of odometry information, which is quite accurate at such small distances. In Figure 3 a histogram is plotted of the difference between estimated and the ground truth heading. The standard deviation of the error is 0.31 radials. Although the images were less than 3 meters apart, the results indicate what we can expect of the heading estimation during navigation.

B. Appearance based mapping

The robot was driven manually through the environment, consisting of a U-shaped hallway and 3 rooms. While the robot was driving images were taken at a rate of 1 image per second. In Figure 4 the approximate positions of the images are shown. The position of these images were derived using the odometry information of the robot. Errors in the odometry were corrected somewhat to make the visualization more clear. This is also used for the figure showing the graph. We must stress that the odometry errors did not influence the outcome of the navigation, as we do not use the odometry readings in our methods.

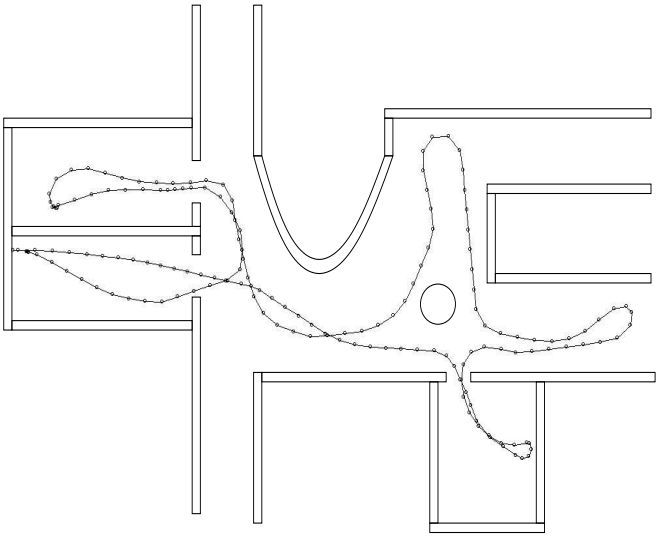

Fig. 4. The path the robot followed while it was manually driven through the environment. The robot started at the lower left of the figure and drove towards the room on the lower right. The circles denote the positions on which panoramic images were taken.

An appearance based topological map is constructed using the images as described in section II, see Figure 5 for the result. The value that is used to threshold the similarity value, is set to 0.05, which seemed to work well for a different dataset taken in another environment. This basically means that 5 out of the 100 image features should have a corresponding feature in the other image, which is constrained by the epipolar geometry.

As can be seen in the figure no links were created between images that were taken from very different locations. The matching method thus shows to be robust against similar looking but different office rooms.

C. Robot navigation

The robot is put on a position in the mapped environment and a goal node is picked from the graph in another part of the environment. This is repeated two times creating two start and end positions for which the robot should find a path and navigate over it. We let the robot navigate 3 times over both paths.

All 6 runs were completed successfully, without having to use the recovery method. In two occasions a heading to the subgoal could not be calculated. However this did not cause the robot to loose track of his path. The robot drove smoothly to the goal node stopping in its vicinity. In Figure 6 two of the traversed paths are shown. The other 4 paths were very similar to these ones. As can be seen the robot did not drive a the trajectory that was driven while taking the dataset. Rather, it used a path of nodes that was much shorter.

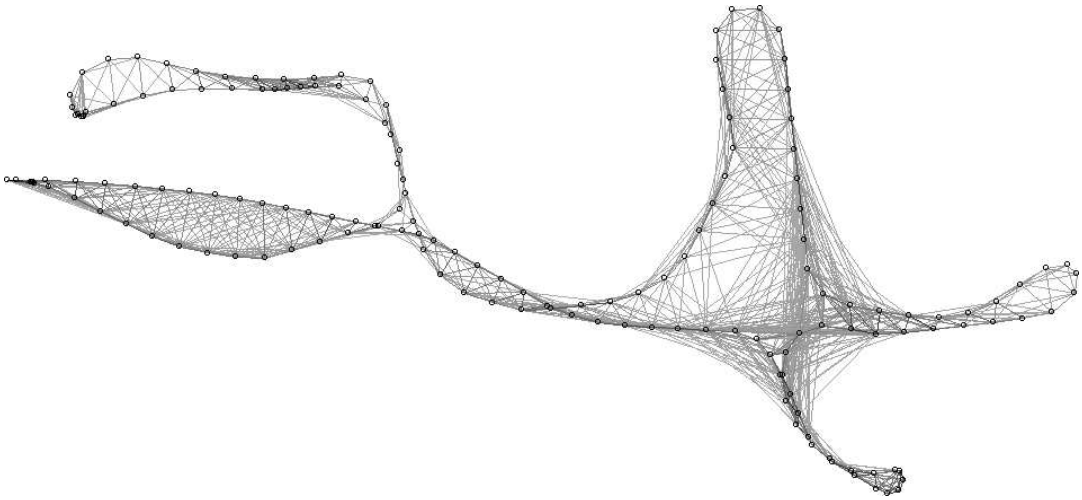

Fig. 5. The appearance based graph. The circles again denote the approximate image positions and the lines connecting them indicate matching images. The gray-value of the lines correspond with the similarity value of the match.

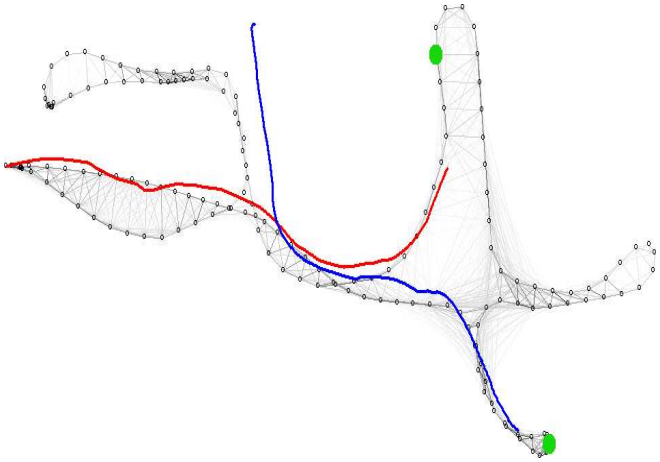

Fig. 6. Two of the traversed paths depicted by the thick blue and red line visualized on top of the graph. As can be seen the paths were quite smooth. The fact that the upper left part of the blue path does not lie on the graph is probably a result of bad odometry readings.

TABLE I

AVERAGE DRIVEN PATH LENGTHS IN METERS $\pm$ THE STANDARD DEVIATION FOR AUTONOMOUS AND MANUAL NAVIGATION

	path 1	**path 2**
Auto	$13.8 \pm .4$	$12.4 \pm .8$
Manual	$14.2 \pm .3$	$12.1 \pm .3$

TABLE II

PATH LENGTHS IN METERS WITH PEOPLE BLOCKING THE VIEW

#Persons	**path lengths**
0	14.2
1	15.2
2	18.0
3	19.0
4	23.6

Quantitatively evaluating the performance of robot navigation is not a straightforward task. It is common to report the metric error of the final robot position given an exact goal position [10], [9]. However this error depends solely on the last stages of the navigation task, which is only interesting if the start and goal position lie close together. It seems more important to measure if the robot "takes wrong turns" while driving through the environment, from which it has to backup. This would have a great impact on the length of the path the robot traversed. In table I the average path length and the standard deviation is given. For comparison the robot was also driven manually from the start positions to the positions where the robot had stopped, by an experienced user. This is also repeated 3 times per path. The lengths of the manually driven paths are comparable with those of the autonomously driven paths, indicating that the robot did follow a correct path to the goal position (see table I).

D. Navigation with visual occlusions

To put more strain on the visual navigation method we now test the ability to drive while part of the view is blocked by people walking next to and in front of the robot. See Figure 7 for an indication of the view the robot has while 4 persons are standing next to it. The persons are walking very near the robot at more or less 20 cm distance. The path that had to be traversed is the same as one of the paths in the previous section. Tests are conducted with respectively 1, 2, 3 and 4 persons.

The robot still managed to reach the goal location in all 4 tests. Nonetheless it was clear that for every person that was added, the navigation was a bit more difficult. In table II it is shown that the path length increases if a larger part of the view is blocked. This is not only caused by small divergences of the correct path, but also because the robot sometimes took a longer route around the pillar in the hallway. Surprisingly the robot never had to use its recovery method. The number of times that the heading to the subgoal

Fig. 7. Four persons blocking the view of the robot.

could not be estimated did increase though. For one and two persons it could still match 100% of the observations, but this decreased to 90% for the runs with three person and four persons.

During the test with 4 persons an additional thing happened. Because no collision avoidance was used and the robot was sometimes heading for a doorpost or the pillar, we had to stop it manually and push it back. This happened 3 times.

VI. Conclusion

We presented a navigation system that can use an appearance based topological map as its representation of the environment. The robot is able to find and traverse paths in the visual domain and can navigate from one state to the other. The found paths do not differ significantly with paths taken by a human, when comparing the path lengths.

Navigation proved to be robust in a dynamic environment with people walking close to the robot. Our navigation system is based on a search for a path given all the images available. This in contrast to existing systems for visual navigation, which drive over predefined paths of images, which were learnt during the exploration phase or even hard coded in a map given to the robot. Our approach is robust against changes in the environment and persons or objects blocking certain paths.

Note that our navigation system is based only on an omnidirectional vision system. This same sensor can be used for a range of other tasks such as object or person detection. Also the appearance based map we use for navigation, is used in other work for localization and conceptualization, splitting the map into rooms, corridors, etc.

Currently we are integrating the navigation approach in a more complete robot system that incorporates people detection, people following and exploration. All these methods make use of the same omnidirectional vision system.

VII. Acknowledgment

The work described in this paper was conducted within the EU Integrated Project COGNIRON ("The Cognitive Companion") and was funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020.

References

[1] C. Stachniss, O. Martínez-Mozos, A. Rottmann, and W. Burgard, "Semantic labeling of places," in *Proceedings of the International Symposium on Robotics Research*, San Francisco, CA, USA, 2005.

[2] A. Tapus and R. Siegwart, "A cognitive modeling of space using fingerprints of places for mobile robot navigation," in *IEEE Conf. on Robotics and Automation*, Orlando, Florida, May 2006.

[3] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," in *Proceedings of ICRA 2000*, vol. 2, April 2000, pp. 1023 – 1029.

[4] S. Vasudevan, S. Gachter, M. Berger, and R. Siegwart, "Cognitive maps for mobile robots an object based approach," in *Proceedings of the IEEE IROS 2006 workshop - From Sensors to Human Spatial Concepts (FS2HSC 2006)*, Beijing, China, 2006.

[5] S. Se, D. Lowe, and J. Little, "Global localization using distinctive visual features," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, October 2002, pp. 226–231. [Online]. Available: citeseer.ist.psu.edu/article/se02global.html

[6] Z. Zivkovic, B. Bakker, and B. Kröse, "Hierarchical map building using visual landmarks and geometric constraints," in *Intl. Conf. on Intelligent Robotics and Systems*. Edmundton, Canada: IEEE/JRS, August 2005.

[7] B. Bakker, Z. Zivkovic, and B. Kröse, "Hierarchical dynamic programming for robot path planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 3720–3725.

[8] Z. Zivkovic, B. Bakker, and B. Kröse, "Hierarchical map building and planning based on graph partitioning," in *IEEE International Conference on Robotics and Automation*, 2006, pp. 803–809.

[9] G. Mariottini, G. Oriolo, and D. Prattichizzo, "Image-based visual servoing for nonholonomic mobile robots with central catadioptric camera," in *ICRA*, Orlando, Florida, May, 15-19 2006, pp. 497 – 503.

[10] A. A. Argyros, K. E. Bekris, S. C. Orfanoudakis, and L. E. Kavraki, "Robot homing by exploiting panoramic vision," *Autonomous Robots*, vol. 19, no. 1, pp. 7–25, 2005.

[11] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?"," in *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, vol. 1. Springer-Verlag, 2002, pp. 414–431.

[12] Z. Zivkovic and O. Booij, "How did we built our hyperbolic mirror omnidirectional camera - practical issues and basic geometry," University of Amsterdam, Tech. Rep. IAS-UVA-05-04, 2005.

[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[14] T. Spexard, S. Li, B. Wrede, J. Fritsch, G. Sagerer, O. Booij, Z. Zivkovic, B. Terwijn, and B. Kröse, "Biron, where are you? - enabling a robot to learn new places in a real home environment by integrating spoken dialog and visual localization," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. IEEE, October 2006, to appear.

[15] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision, secon edition*. Cambridge University Press, 2003.

[16] P. H. S. Torr and D. W. Murray, "The development and comparison of robust methods for estimating the fundamental matrix," *IJCV*, vol. 24, no. 3, pp. 271–300, 1997.

[17] M. Brooks, L. de Agapito, D. Huynh, and L. Baumela, "Towards robust metric reconstruction via a dynamic uncalibrated stereo head," 1998. [Online]. Available: citeseer.csail.mit.edu/brooks98towards.html

[18] J. Kosecká, F. Li, and X. Yang, "Global localization and relative positioning based on scale-invariant keypoints." *Robotics and Autonomous Systems*, vol. 52, no. 1, pp. 27–38, 2005.

[19] B. K. P. Horn, "Relative orientation," *Int. J. Comput. Vision*, vol. 4, no. 1, pp. 59–78, 1990.

[20] O. Chum, T. Werner, and J. Matas, "Epipolar geometry estimation via ransac benefits from the oriented epipolar constraint," *icpr*, vol. 01, pp. 112–115, 2004.

[21] Y. Mezouar, H. Hadj Abdelkader, P. Martinet, and F. Chaumette, "Visual servoing from 3d straight lines with central catadioptric cameras," in *Fifth Workshop on Omnidirectional Vision, Omnivis'2004*, Prague, Czech Republic, May 2004.

[22] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, December 1959.