Image based navigation using a topological map

O. Booij, B. Terwijn, Z. Zivkovic, B. Kröse

Intelligent Systems Lab Amsterdam
University of Amterdam
1098 SM Amsterdam, The Netherlands
obooij@science.uva.nl

Keywords: Vision based navigation, topological maps, robot navigation

Abstract

A new approach on robot navigation is described which enables the robot to drive through the environment using only a vision system as its sensor. In an exploration phase the robot is driven manually through the environment while taking images. By computing similarities between these images the robot constructs a topological map in the form of an appearance graph. Navigation on this graph involves homing from one node to the other until the goal node is reached. No explicit metrical information is modelled by this approach. Rather the distances in the topological map provide directly information about the ability and the robustness of visual navigation in the environment. Real world experiments were conducted under varying environmental conditions.

1 Introduction

Traditionally robots are designed for and deployed in controlled environments such as factory buildings and assembly lines. Today we see a growing interest in designing robots that can operate in real home environments and interact with people through speech and gestures. These 'personal' robots have quite different requirements which create a demand for different approaches to common robot tasks. In contrast to factories, the environmental conditions in real homes are dynamic and hard to predict.

A personal robot should be able to interact with humans which requires the robot to detect humans, recognize them and follow them. Also the robot should be able to learn to recognize new objects and rooms. Computer vision is well suited to accomplish all of these tasks. It would be beneficial to perform all other more traditional robot tasks such as localization and navigation also on the basis of vision in spite of the good results obtained by other sensors for these specific tasks such as the laser range finder. Another, more pragmatic, reason to use a vision system as the main sensor are the low costs, which is important if one aims at designing a consumer product.

Another requirement is that a personal robot should be able to navigate in a house without being provided with a ground floor map. Thus it should be able to acquire a spatial representation using the vision system by itself.

In this paper we present how an appearance based (also known as view based) representation of the environment can be used for goal directed visual navigation. This method builds on existing work of building a so called appearance graph[1], which already proved to be useful for localization[2], hierarchical path-planning[3] and for finding convex spaces by partitioning the graph[4].

This paper is structured as follows. First we give a brief overview of related work on visual navigation in Section 2. Then in Section 3 we give a concise description on how an appearance based graph is build. Section 4 presents the method that is used to estimate the relative pose between two images by computing the epipolar geometry. This estimation method is then used in the navigation system to navigate over the graph as explained in Section 5. Real world experiments were conducted with different levels of occlusion which are reported in Section 6. And finally we draw some conclusions and give pointers to future work in Section 7.

2 Related work

The state of the art in map building and localization, which are both important steps toward successful navigation, is EKF-SLAM, i.e. the Extended Kalman Filter solution of the Simultaneous Localization And Mapping problem [5]. In general this is applied on data acquired by a laser range scanner and the odometry of the robot. However also vision based solutions have been presented [6, 7]. The resulting spatial representation is a 3D map, which makes path planning and navigation relatively straight-forward. The difficulty of such solutions is that the 3D positions of the landmarks are used while the camera only provides bearing information. Also the number of landmarks rapidly grows when the environment becomes larger, making it impossible to maintain a consistent state and covariance estimate.

Appearance based or view based approaches, as the one proposed in this work, avoid this problem, because they only model relations between the images taken from the environment[8, 9]. The number of images is in general much smaller than the number of landmarks. Navigation using such an appearance based map is performed by driving from one image to the other until the goal image is reached. This is commonly called visual homing or visual servoing. A lot of work in this field is inspired by the biological mechanisms of nest-finding, which resulted in behavioristic methods that use simple heuristics to navigate to an image [10, 11, 12]. Most, if not all, of this work has only been tested in small environments.

More principled approaches were taken by [13, 14], in which the homing direction was estimated using the local geometry between the images. However again only small scale experiments are reported. Our method also uses the epipolar geometry estimation as explained in [13]. However we make it more robust by incorporating the fact that the robot drives over a plane ground-floor and that points in the images can only be produced by landmarks lying in front of the camera.

A difference between all these approaches and the work described in this paper is the manner in which a path, a sequence of images, is planned in the appearance based map. Existing approaches use sequences of images as shot during the exploration phase. Not only does this prohibit them to take short-cuts, but is also error prone under changing environmental conditions. An exception is the method described in [15]. However that method can only navigate a robot in a single room.

In this paper we describe a navigation system that actively searches for a path while it is driving to the goal node. This is useful in dynamic environments where persons might block the path of the robot and light conditions are constantly changing.

3 Building an appearance graph

In this section we describe the method used for constructing the appearance based map. This has already been reported in [1]. 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 [16]. 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.

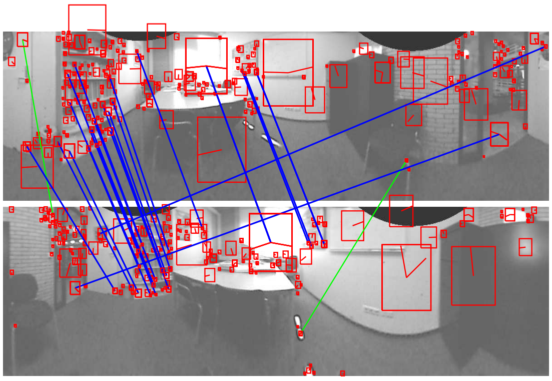

Figure 1: Matching two images. The red boxes indicate the SIFT features found in the images. The lines connecting two of these features indicate that they correspond. If the line is blue, this means the corresponding pair agrees with the epipolar constraint. If it is green, it does not agree with the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

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 [17] for details). From each of the resulting panoramic images a set of SIFT features is extracted [18]. 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 4). 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 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 4 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 [1] 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 [19].

4 Heading estimation using the epipolar geometry

Epipolar geometry estimation is thoroughly discussed in computer vision literature and some standard implementations are readily available [20, 21]. 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 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 [20]. 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, leaving only 4 free parameters [22, 23]:

$$E = \begin{bmatrix} 0 & e_2 & 0 \\ e_4 & 0 & e_6 \\ 0 & e_8 & 0 \end{bmatrix}, \tag{2}$$

The minimal number of point pairs for the linear estimation of this constrained essential matrix is only 3 and because the solution space is smaller the estimator is less effected by noise.

The essential matrix bears the relative rotation R and translation $\mathbf{t}$ 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 $\mathbf{t}$. The essential matrix can be decomposed into 4 different solutions of $\mathbf{t}$ 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 [24]. The world points that were projected behind one of the image surfaces given the correct R and $\mathbf{t}$ 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. 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 [20, 21]. 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 [25]). 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 $\mathbf{t}$ (up to an unknown scale factor) are determined between the image locations[24]. In the remainder of the paper we assume that, if R and $\mathbf{t}$ 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

$$\phi = \text{atan2}(\mathbf{t}_y, \mathbf{t}_x). \tag{4}$$

5 Navigation over the appearance graph

In this section we describe the framework for navigating to a goal location in the environment mapped by the appearance based graph given the heading estimation explained in the previous section. The robot should be able to navigate to any location in the mapped environment by giving it a node in the graph.

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 require us to make an assumption on the position of landmarks in the world, making the system less flexible. Another solution is to track features in three or more images, which is quite common in the field of visual servoing (see for example [26]). However, in environments where humans frequently block the view of the robot it is difficult to track image points robustly.

We take it that the goal location is given by a node in the graph. First Dijkstra's shortest path algorithm [27] 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 a distance as $d_{ij} = \frac{1}{S_{ij}}$. The distances of the nodes to goal node are used during driving as a heuristic to find nodes closer to the goal node.

The navigation procedure directs the robot to one node at a time which 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 3). The node of the graph with the highest similarity is chosen as the current subgoal node c. This procedure is linear in the number of nodes and could thus be time consuming, but is only required at the start of navigation.

After a subgoal node is determined the robot tries to pick a better 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 this new c, until the node is found that is closest to the goal node and still robustly matches with the new observation, given the same threshold as used for constructing the graph.

When a subgoal is determined, the heading is estimated in order to drive in its direction using equation (4). 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 takes previous heading estimates into account.

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 3 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 node as subgoal. See Algorithm 1 for an overview of the navigation method.

Algorithm 1 Navigation over the appearance graph

```
repeat
  take new image $i$
  if $c$ = UNDEFINED then
    compute $s_{ij}$ for all $j$
    $c \Leftarrow \mathrm{argmax}_j \, s_{ij}$ { global localization }
  end if
  repeat
    for all neighbors $k$ of $c$ with $D_k < D_c$ do
      compute $s_{ik}$
      if $s_{ik}$ < THRESHOLD then
        $newc \Leftarrow k$ {new current subgoal}
      end if
    end for
    $c \Leftarrow newc$
  until $c$ did not change
  compute heading $\phi$ from image $i$ to $c$
  $\theta \Leftarrow$ recency weighted averaging filter($\theta$, $\phi$)
  Move robot in direction $\theta$
until $c$ = GOAL
```

6 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 heading estimation and navigation procedure are tested in an office environment.

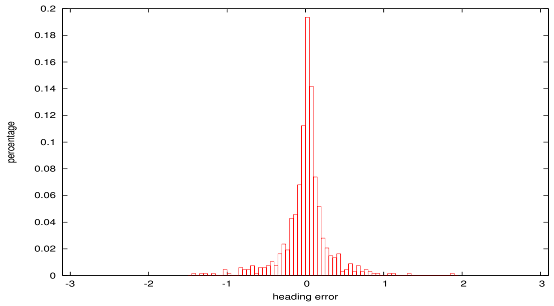

Figure 2: Histogram of the differences between the estimated and the ground truth heading for all pairs of images.

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.

6.1 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 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, resulting in a set of 36 images.

For each pair of images the heading is computed given the method explained in sections 3 and 4. 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 2 a histogram is plotted of the difference between estimated and the ground truth heading. The standard deviation of the error is 0.31 radials.

6.2 Appearance based mapping

The robot was driven manually through the environment, consisting of a U-shaped hallway and 3 rooms, see Figure 3. While the robot was driving images were taken at a rate of 1 Hz. In Figure 3 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 done for other figures 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.

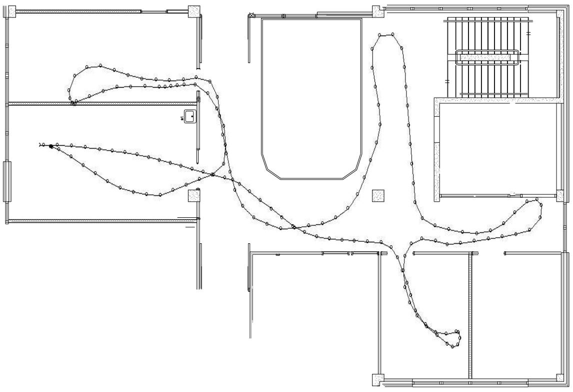

Figure 3: The path the robot followed while it was manually driven through the environment. The robot started at the lower left of the figure and drove towards the room on the lower right. The circles denote the positions on which panoramic images were taken.

An appearance graph is constructed using the images as described in section 3, see Figure 4 for the result. The value that is used to threshold the similarity value, is set to 0.05, which was chosen because it seemed to work well for a different dataset taken in another environment. This basically means that at least 5 out of a 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.

6.3 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 a subgoal node 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 5 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 trajectory that was driven while taking the dataset. Rather, it used a path of nodes that was much shorter.

Quantitatively evaluating the performance of robot navigation is not a straightforward task. It is common

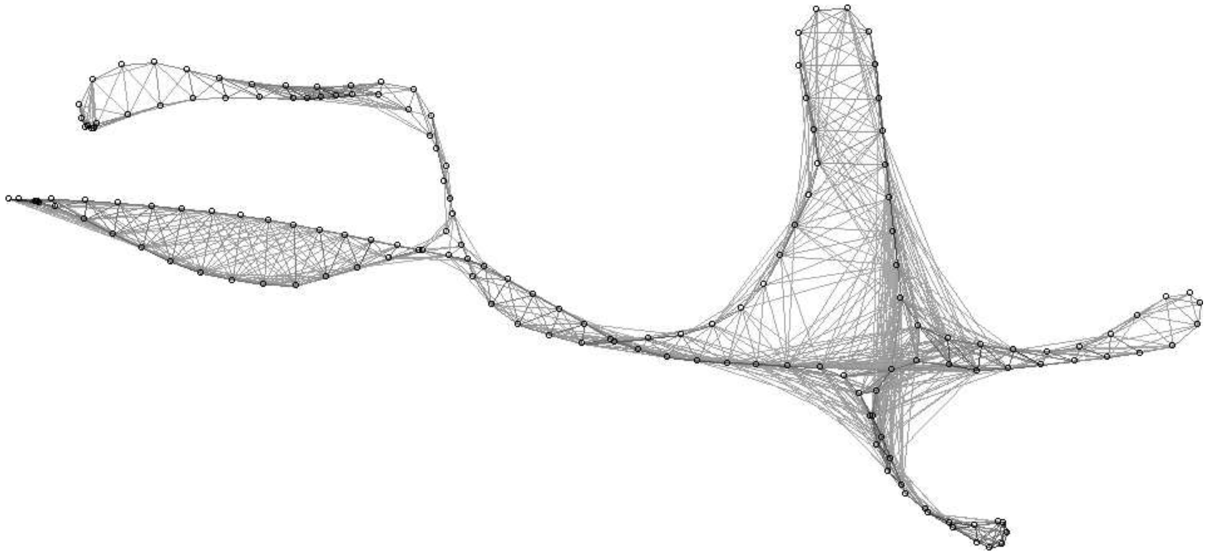

Figure 4: The appearance based graph. The circles again denote the approximate image positions and the lines connecting them indicate matching images. The gray-value of the lines correspond with the similarity value of the match.

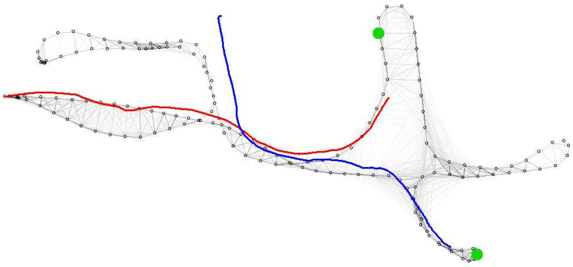

Figure 5: Two of the traversed paths depicted by the thick blue and red line visualized on top of the graph starting at the red circles and ending at the filled green circles. As can be seen the paths were quite smooth. The fact that the upper left part of the blue path does not lie on the graph is the result of bad odometry readings.

Table 1: Average driven path lengths in meters ± the standard deviation for autonomous and manual navigation.

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

to report the metric error of the final robot position given an exact goal position [10, 13]. 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 recover. This would have a great impact on the length of the path the robot traversed. In Table 1 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 1).

6.4 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 6 for an indication of the view the robot has while 4 persons are standing next to it.

Table 2: 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

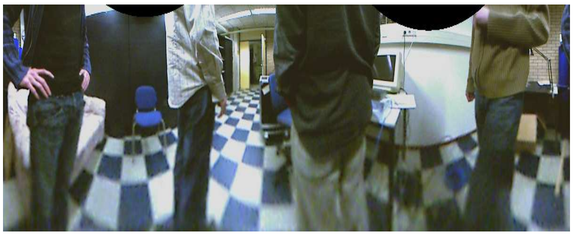

Figure 6: Four persons blocking the view of the robot.

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 2 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 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 the robot was sometimes heading for a doorpost or the pillar. Because no collision avoidance was used, we had to stop it manually and push it back. This situation occurred 3 times.
.

7 Conclusion

In this paper we proposed a novel method for vision based navigation which uses an appearance based topological map of the environment. The distances contained in the map do not explicitly correspond to metric distances in the real world but give an indication of the ability to travel a certain path. Thus while the robot is searching for the shortest path in the appearance graph it is searching for the path in the real world which it can traverse with the least difficulty.

Even with a high degree of occlusion, caused by people walking close to the robot, it was able to navigate to the goal position. Interestingly the robot took a different path when 4 persons were blocking its view, because that path was shortest in appearance space under those particular circumstances. This shows the advantage of searching for the best path while driving, in contrast to existing approaches that travel a predefined path. In the near future this will be further tested using other types of challenges, like closing doors and switching of part of the lights. Also tests should be conducted in still larger environments.

Note that the shortest path in the appearance space 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.

8 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] 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, August 2005. IEEE/JRS.

[2] O. Booij, Z. Zivkovic, and B. Kröse. Sparse appearance based modeling for robot localization. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1510–1515. IEEE, October 2006.

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

[4] Z. Zivkovic, O. Booij, and B. Kröse. From images to rooms. *Robotic and Autonomous Systems*, 2007. to appear.

[5] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localisation and mapping (slam): Part i the essential algorithms. *Robotics and Automation Magazine*, June 2006.

[6] N. Karlsson, E. Di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich. The vSLAM algorithm for robust localization and mapping. In *2005 IEEE International Conf. on Robotics and Automation, ICRA 2005*, 2005.

[7] S. Se, D. G. Lowe, and J. J. Little. Vision-Based Global Localization and Mapping for Mobile Robots. *IEEE Transactions on Robotics*, 21(3):364–375, 2005.

[8] B. J. A. Kröse, N. A. Vlassis, R. Bunschoten, and Y. Motomura. A probabilistic model for appearance-based robot localization. *Image and Vision Computing*, 19(6):381–391, April 2001.

[9] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *Proceedings of ICRA 2000*, volume 2, pages 1023 – 1029, April 2000.

[10] A. A. Argyros, K. E. Bekris, S. C. Orfanoudakis, and L. E. Kavraki. Robot homing by exploiting panoramic vision. *Autonomous Robots*, 19(1):7–25, 2005.

[11] M.O. Franz, B. Schlkopf, and H.H. Blthoff. Where did i take that snapshot? scene-based homing by image matching. *Biological Cybernetics*, 79:191–202, 1998.

[12] Y. Matsutmoto, K. Ikeda, M. Inaba, and H. Inoue. Visual navigation using omnidirectional view sequence. In *Proc. of 1999 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pages 317–322, Kyongju, Korea, Oct 1999.

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

[14] G. Blanc, Y. Mezouar, and P. Martinet. Indoor navigation of a wheeled mobile robot along visual routes. In *Proceedings of ICRA 2005*, pages 3365–3370, Barcelona, Spain, April 2005. IEEE.

[15] Emanuele Menegatti, Takeshi Maeda, and Hiroshi Ishiguro. Image-based memory for robot navigation using properties of omnidirectional images. *Robotics and Autonomous Systems*, 47(4):251–267, 2004.

[16] 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*, volume 1, pages 414–431. Springer-Verlag, 2002.

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

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

[19] Thorsten Spexard, Shuyin Li, Britta Wrede, Jannik Fritsch, Gerhard Sagerer, Olaf Booij, Zoran Zivkovic, Bas Terwijn, and Ben 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.

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

[21] P. H. S. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *IJCV*, 24(3):271–300, 1997.

[22] M. Brooks, L. de Agapito, D. Huynh, and L. Baumela. Towards robust metric reconstruction via a dynamic uncalibrated stereo head, 1998.

[23] Jana Kosecká, Fayin Li, and Xialong Yang. Global localization and relative positioning based on scale-invariant keypoints. *Robotics and Autonomous Systems*, 52(1):27–38, 2005.

[24] B. K. P. Horn. Relative orientation. *Int. J. Comput. Vision*, 4(1):59–78, 1990.

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

[26] 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.

[27] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, December 1959.