Hierarchical Map Building Using Visual Landmarks and Geometric Constraints *

Zoran Zivkovic and Bram Bakker and Ben Kröse
Intelligent Autonomous Systems Group
University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
{*zivkovic,bram,krose*}*@science.uva.nl*

***Abstract*— This paper addresses the problem of automatic construction of a hierarchical map from images. Our approach departs from a large collection of omnidirectional images taken at many locations in a building. First a low-level map is built that consists of a graph in which relations between images are represented. For this we use a metric based on visual landmarks (SIFT features) and geometrical constraints. Then we use a graph partitioning method to cluster nodes and in this way construct the high-level map. Experiments on real data show that meaningful higher and lower level maps are obtained, which can be used for accurate localization and planning.**

***Index Terms*— mobile robots, vision based navigation, hierarchical map building, topological map**

I. INTRODUCTION

Mobile robot localization and navigation requires an internal representation of the environment. Traditionally such a model is represented as a 2D geometric model of the workspace of the robot, indicating admissible and non-admissible areas. Because of recent progress in sensor technology (vision sensors, 2D laser scanners), these models tend to become quite complex (such as 3D planar maps with texture, 3D landmark positions), resulting in a large number of parameters that have to be stored and estimated.

Hierarchal approaches combining higher level conceptual maps (usually topological maps - graph structures with nodes representing places and edges or links representing possible transitions) with lower-level, geometrically accurate, local maps have a number of advantages. One of the problems with complex (3D) maps is that the number of parameters which have to be estimated in a SLAM procedure increases very fast with the spatial extent of the map. The advantage of splitting the representation into smaller parts is that it makes better parameter estimation possible, if the new problem of maintaining consistency between local representations can be solved [12]. A second advantage of a hierarchical representation is that hierarchical path planning methods can be used. We show in [3] that such planning methods have computational advantages over non-hierarchical planning methods. Finally, a third advantage of a hierarchy of maps is that it can facilitate the interaction of the robot with humans, because the elements in the higher-level map (e.g., the nodes in the graph) can be made to correspond to concepts that make sense to humans (rooms, corridors), instead of metric (x,y) coordinates that are not intrinsically meaningful to humans in office and home environments.

The issue addressed in this paper is how to create a higher level conceptual map which can be used in a hierarchical framework. Different approaches have been proposed in earlier work. In human augmented mapping, a human supervisor indicates which places are to serve as nodes in the graph [2]. Alternatively, an existing metric representation can be used to derive a higher level topological map using geometrical methods such as generalized Voronoi graphs [5]. It is also possible to use sensory data directly for the creation of a higher level map. In [16], [10] a set of images of the robot's environment is grouped based on the presence of a number of automatically extracted landmarks.

In this paper we describe an alternative algorithm for generating a higher level topological map directly from images. The algorithm is based on an appearance-based representation, which is a representation where the environment is not modelled geometrically, but as an 'appearance map' that consists of a collection of sensor readings obtained at various poses [11],[14],[8]. In our approach we do not assume the poses to be known but just use an unordered collection of omnidirectional images taken at many places in the building. The algorithm first constructs a graph ('low level' topological map) from all images by using a grouping criterion that takes into account both the presence of the visual landmarks (SIFT features) and the constraints imposed by the geometry of the environment. This initial step is similar to an algorithm from a different field [21] that was used to group images

*The work described in this paper was conducted within the EU FP6-002020 COGNIRON ("The Cognitive Companion") project. We would also like to thank Olaf Booij for useful comments.

from the same scene of a TV movie. We further define a criterion for grouping the images of the environment so that images from a convex area, a room for example, are naturally grouped together. This criterion corresponds to the normalized graph cut criteria from graph theory [9]. The exact solution is computationally expensive and we use a standard approximate solution [9].

In section 2 we give a brief overview of space representations in robotics and relate them to the method presented in this paper. In section 3 we describe how to generate a low-level topological map of the space from the images of the environment. Section 4 gives some details about the underlying computer vision algorithms needed to do that. Section 5 describes our graph-theoretic method for grouping the images and extracting a higher level conceptual map of the space from the low-level topological map defined in sections 3 and 4. Our experimental results are presented and discussed in Section 6.

II. Related Work - Higher Level (Topological) Maps From Images

Our method is an appearance map method based on images obtained by the robot using a visual sensor. The images are 2D projections of the 3D space. Standard algorithms for 3D reconstruction from images [7] usually extract the metric information incrementally. Typically the different levels of extracted metric information are:

Step 1: At this level images are grouped based solely on their immediate appearance. Images from the same part of the environment (a room, a corridor) are expected to look similar. Typically, images are grouped together by determining whether they have similar landmarks. This grouping based on immediate image appearance produces a straightforward higher level representation of space. However, in large environments the probability that images from completely different places are grouped together can become high. This is sometimes called 'perceptual aliasing'. To reduce perceptual aliasing, [16] proposes to take into account the horizontal ordering of landmarks in the image, yielding what they call 'fingerprint representations'. Similarly, [10] proposes a global description of images using SIFT features [13] as landmarks and their distribution within the image. This provided a more distinctive representation of the space.

Step 2: At this level the images are grouped based on their immediate appearance but also on the geometry of the space. From two images and a set of matching landmarks one can perform two-view geometric reconstruction of the space (see [7] and section IV). This requires that not only similar landmarks are present, but also that they come from the same real-world 3D positions (up to a scale factor). This requirement is much stricter than those in [16],[10]. Therefore, perceptual aliasing is very rare even in large environments (see [21] and also the remainder of this paper).

Step 3: By matching the landmarks over more than two views it is possible to reconstruct the camera poses for the images, 3D positions of the landmarks, and finally perform dense 3D reconstruction of the space (up to a scale factor) [7]. This dense 3D reconstruction can then be used to obtain a precise 2D geometrical map of the environment. A higher level conceptual map can, in turn, be extracted from this 2D geometrical map using the methods described in [23],[15]. Note that to apply these methods we need to use complex and computationally expensive algorithms to perform the complete 3D reconstruction, and that in general the 3D reconstruction problem cannot be considered completely solved, especially for large environments.

The metric reconstruction in this paper stops at step 2, where geometric constraints are imposed, and for example information about occlusions and visibility is already present. In the remainder of this paper we show (sections 3-5) how to use this information and build a natural higher-level representation of the space.

III. Lower Level Topological Map From Images Using Appearance and Geometrical Constraints

A general definition of a topological map is that it is a graph-like representation of space. A set of n nodes V of the graph represent distinct positions in space, and edges or links of the graph encode how to navigate from one node to the other [6]. The nodes and the edges can be enriched with some local metric information.

In this paper, as is typical in appearance-based approaches, each node represents a location and corresponds to an image taken at that location. As the result from n images we get a graph with n nodes that is described by a symmetric matrix S called the 'similarity matrix'. For each pair of nodes $i,j \epsilon [1,...,n]$ the value of the element S_{ij} from S defines the similarity of the nodes. In our approach S_{ij} is equal to 1 if and only if it is possible to perform 3D reconstruction of the local space from the two images corresponding to the nodes. Otherwise there is no link between the nodes and $S_{ij} = 0$. Our 3D reconstruction is based on the Scale Invariant Feature Transform (SIFT) features [13] as the automatically detected landmarks in the image. The algorithm we are using for the 3D reconstruction is described in more detail in section IV. An example of such a graph that we obtained from a real data set is given in figure 2b.

This graph contains, in a natural way, information about how the space in an indoor environment is separated by the walls and other barriers. Images from a convex space, for example a room, will have many connections between them, and just a few connections to images from another convex

space, for example a corridor, that is connected with the room via a narrow passage, for example a door. In section 5 we describe how to extract such groups of images automatically from the graph (V, S).

There are various ways to define the similarity metric for S_{ij}. The simple metric we use is directly related to the robot navigation task. For localization and navigation the robot can use the same algorithm as the one used to define the edges of the graph (V, S). An edge in the graph denotes that 3D reconstruction is possible between the images that correspond to the nodes. This also means that if the robot is at one node it can determine the relative location of the other node. Therefore, if there are no obstacles in between, the robot can directly navigate from one node to the other (as, e.g., in [17]). If there are obstacles, one could rely, for example, on an additional reactive algorithm for obstacle avoidance that is using range sensors. Furthermore, additional information can be associated with the edges of the graph. For example, if we reconstruct the metric positions of the nodes (using the images or if we measure them in some other way), we could also associate the Euclidean distance between the nodes with each edge. This could be used for better navigation and path planning using the graph. However, this is beyond the scope of this paper.

IV. Visual Landmarks and Geometric Constraints

Having described the general process of constructing the lower level topological map, we proceed to describe some of the details of the underlying computer vision algorithms. First we extract distinctive points from images. Examples are a corner, T-junction, a white dot on black background etc. Such points are often used in the computer vision community as automatically detected landmarks. Here we use the SIFT feature detector [13]. The SIFT feature detector extracts also the scale of the feature point and describes the local neighborhood of the point by a 128-element rotation and scale invariant vector. This vector descriptor is also robust to some light changes.

A. Matching Landmarks

Visual landmarks are used often in robotics for navigation [22],[19],[18]. It is possible to reconstruct both the camera images and the 3D positions of the landmarks by matching (or tracking) landmarks through images. On-line simultaneous localization and reconstruction of landmark positions was presented in [1], but currently only for small scale environments.

In this paper we consider the general case when we start with a set of unordered images of the environment. This is similar to [20]. In practice we often have some information about ordering of the images (e.g. a movie as in [1]) or some other sensor readings (odometry for example), which should be used in that case.

Most 3D reconstruction algorithms [7] start with finding similar landmarks in pairs of images. When two images are consecutive frames of an image sequence we could track the landmarks from one image to the other [1]. However, it is much more difficult to find matching landmarks in an unordered set of images. Firstly, we need to check all the pairs of images, which is computationally expensive. Secondly, there are no additional constraints as is generally the case in an image sequence.

In this paper we use a heuristic similar to [21]. For each landmark from one image we find the best and the second best matching landmark from the second image. The goodness of the match is defined by the Euclidean distance between the landmark descriptors. If the goodness of the second best match is less then 0.8 of the best one it means that the match is very distinctive. According to the experiments in [13], this typically discards 95% of the false matches and less then 5% of the good ones. This is repeated for each pair of images and it is computationally expensive. Fast approximate methods were discussed in [13]. Since our data sets were not very big we performed the full extensive search.

B. Geometric Constraints

The method described in the previous section finds the possible matches for each pair of images from our data set. Let there be N matching landmark points between the images m and l. The 2D image positions of the points in the m-th image in the homogenous coordinates are denoted as $\{\vec{p}_1^m, ..., \vec{p}_N^m\}$. The corresponding points in the l-th image are $\{\vec{p}_1^l, ..., \vec{p}_N^l\}$. If the i-th point belongs to the static scene, then, for a projective camera, the positions are related by:

$$(\vec{p}_i^m)^T F \vec{p}_i^l = 0 \quad (1)$$

where the matrix F is also known as the 'fundamental matrix'. Estimating F is an initial step for 3D space reconstruction from images.

In case there are initially many false matches [21], they must be removed using a method to detect and remove outliers. Standard robust M-estimators are commonly used, which can deal with a limited number of outliers. If there are more outliers, the robust algorithm called RANSAC is commonly used [7]. It was shown [24] that a combination that performs best in many cases is when RANSAC is used first and then the M-estimator. Here, we use the distinctive matches criterion, described above and in [13], which already discards many false matches. In our experiments we observed that the number of false matches is small and it is possible to use the robust M-estimator directly. We used the Huber

M-estimator and the standard 8-point algorithm [7] for estimating the fundamental matrix F.

Residuals of fitting the model (1) to each pair of images [7] are used to calculate the global standard deviation σ_{global}. This standard deviation is used to decide when the fundamental matrix is properly calculated. The σ_{global} is estimated robustly using the maximum absolute difference estimate. The whole procedure, then, is as follows:

- extract SIFT landmarks from all images
- find distinctive matches between each pair of images
- if there are more than 8 matches:
  - estimate the fundamental matrix using the M estimator (could be RANSAC)
  - discard matches that deviate more than $2.5\sigma_{global}$
  - if there are still more than 8 matches, add an edge in the graph - set the similarity between these images to 1.

V. Constructing Higher Level Topological Map Using Graph Cuts

The central idea behind our method to construct the higher level topological map is to cut the lower level topological map (described above) into a number of separate clusters, each of which becomes a higher level node or higher level state. We will start by introducing some graph-theoretic terms. The *degree* of the i-th node of the graph (V, S) is defined as the sum of all the edges that start from that node: $d_i = \sum_j S_{ij}$. For nodes A (where A is subset of V), *volume* is defined as $vol(A) = \sum_i d_i$. $vol(A)$ describes the 'strength' of the interconnections within the subset A. Graph V can be divided into two subsets A and B by cutting a number of edges. The sum of the values of the edges that are cut is called a graph cut:

$$cut(A, B) = \sum_{i \epsilon A, j \epsilon B} S_{ij} \quad (2)$$

One may cut V into a number of clusters by minimizing (2). This would mean that the graph is cut at the weakly connected places, which usually correspond to doors between the rooms or other narrow passages. However, such segmentation criteria often leads to undesirable results. For example, if there is an isolated image connected to the rest of the graph by only one link, then by cutting only this link we minimize (2). To avoid such artifacts we use a *normalized* version. The normalized graph cut separates the graph V into two subsets A and B by minimizing the following criterion:

$$nCut(A, B) = (\frac{1}{vol(A)} + \frac{1}{vol(B)}) cut(A, B). \quad (3)$$

Minimizing this criterion means cutting a minimal number of connections between the two subsets but also choosing subsets with strong interconnections. This criterion naturally groups together images from a convex area, like a room, and makes cuts between areas that are weakly connected. The algorithm is simply applied again to obtain more clusters. Finding the optimal solution is computationally expensive. In this paper we use the fast approximate solution from [9].

The following scenario can give another perspective on the normalized cut criterion. An edge means that the robot might navigate from one node to the other as described in Section III. If we assume that the robot randomly moves from a node to a connected node, it is possible to show [9] that: $nCut(A, B) = P(A \rightarrow B|A) + P(B \rightarrow A|B)$. Here $P(A \rightarrow B|A)$ is the probability of jumping from subset A to subset B if we are already in A and $P(B \rightarrow A|B)$ is the other way around. This means that with this random movement, the segmentation is such that the robot has the lowest probability of moving from one cluster to the other.

In [3] we show how path planning can be done using the resulting hierarchical map (the combination of the lower level and higher level topological map), and we show that planning is actually much more efficient using the hierarchical map, compared to just using the lower level map.

VI. Experiments

The experiments described in this paper were designed to investigate the validity of the method to extract the lower level topological map from the images, and the method to extract the higher level topological map from the lower-level map. The experiments were performed using a robot equipped with an omnidirectional camera with a hyperbolic mirror. Circular images were first transformed to panoramic images. Next, the SIFT features were extracted using the standard method [13].

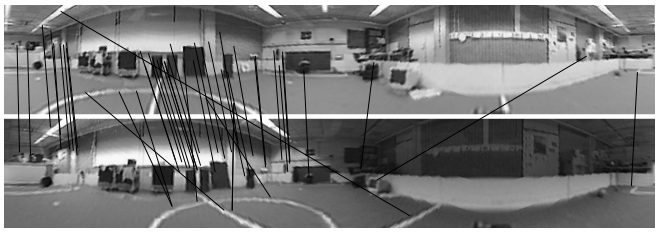
A. Experiment 1: Robustness

Some experimenting was done to test robustness for variability in the images. Figure 1 illustrates the robustness of the method. Despite the different light conditions and occlusions, there were still enough matches to estimate the fundamental matrix. Note that constraint (1) did not remove all false matches. Matches that are false but still close to constraint (1) are not removed.

B. Experiment 2: Perceptual aliasing

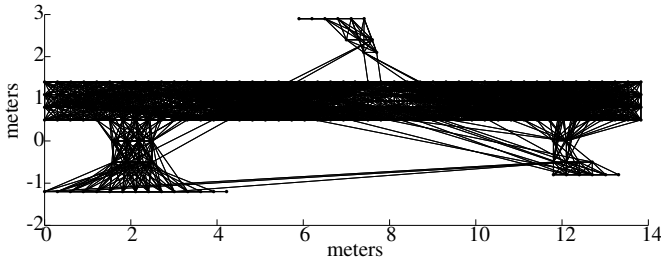
From a data set of 234 images from an office environment we constructed the (lower level) graph using the method described in Section III. The links and the nodes are shown in figure 2. The environment consisted of 3 rooms and a corridor. Two rooms were on one side of the corridor and one on the other side (see also figure 3 which shows the actual layout of the rooms). Figure 2a presents the graph when we match images based only on the presence of the

a) occlusion (the person in the middle of the image)

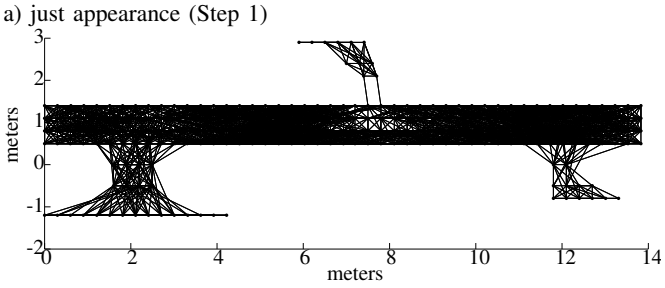


b) different light conditions (two lights turned on or off)

Fig. 1. Pairs of panoramic images taken at approximately 1m distance from each other. The lines indicate the matching landmarks between the two images.



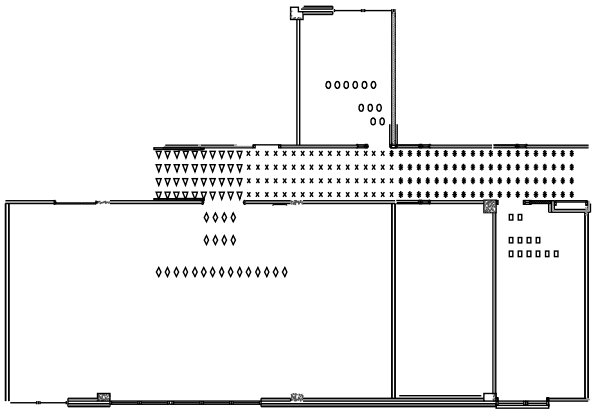
a) just appearance (Step 1)



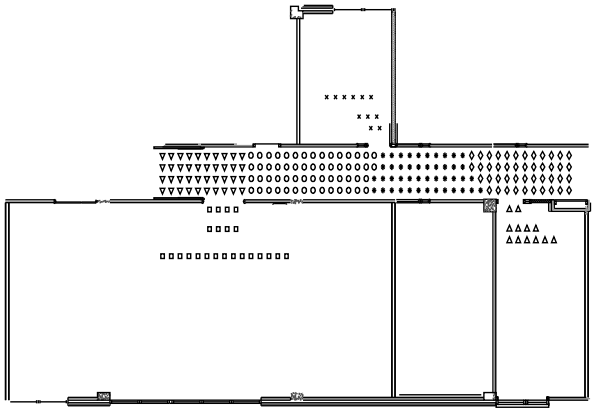
b) with geometric constraints (Step2)

Fig. 2. Reducing perceptual aliasing using geometric constraints. Bird's eye view of the environment with the locations where each of images were taken.

landmarks (see section 2, step 1). The result of taking into account the geometrical constraints is that from the total of 3077 edges, 541 were discarded. For our environment, this removed all perceptual aliasing problems from the graph, as shown in figure 2b.



a) 6 clusters



b) 7 clusters

Fig. 3. Higher level conceptual grouping using minimal normalized cuts. Bird's eye view of the environment with the locations where each of images was taken. Each symbol indicates a specific higher level state. The grouping is obtained directly from the images without using the known ground truth locations.

C. Experiment 3: Building hierarchical map

The normalized graph cut clustering algorithm (section V) was applied to the graph shown in figure 2b. The results, presented in figure 3, show meaningful and natural segmentation of the space. Note that this segmentation was obtained directly from the images in an unsupervised way. One only needs to select the number of clusters, cluster labels need not be assigned to images by a user, and ground truth (actual positions where images were captured) is not used. The results for two different numbers of clusters are shown in the figure.

D. Experiment 4: Higher Level Localization

We applied the algorithm to a data set that contained images that cover a much larger area. Again, a meaningful and natural segmentation of the space is obtained (see figure 4).

For this data set, we perform an experiment to investigate whether the robot can, given a single image it is assumed to observe currently, determine the correct corresponding higher level cluster or higher level state. We select one image from the data set and assume that this is the image observed by the robot. We use the rest of the images as the map. We compute the links of the current image to the images in the map (Section III). The current higher level cluster is then estimated by the robot in a simple way: we look at the cluster labels of the images that have links to the current image and decide the current image's cluster label by a majority vote. This data base has 240 images. For only 5 images the higher level cluster was estimated incorrectly, and they were all at the borders of the clusters.

Note that without any additional information we need to check all the images in order to find the higher level node. It is also possible to speed up this process as discussed in [4].

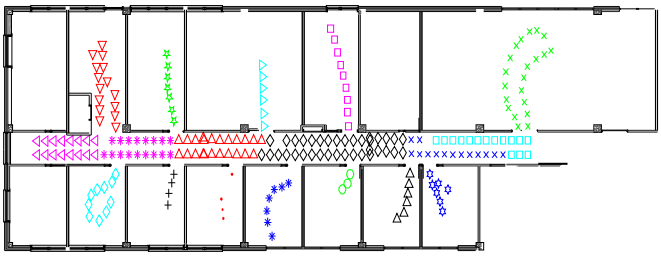


Fig. 4. Higher level conceptual grouping using minimal normalized cuts in a larger environment. Bird's eye view of the environment with the locations where each of the image was taken. Each symbol indicates a specific higher level state. The grouping is obtained directly from the images without using the known ground truth locations.

VII. Conclusions and Further Work

We presented an algorithm for automatically generating hierarchical maps from images. Lower level maps are directly derived from images, higher-level maps are derived from the lower level maps. Experiments on real data show that meaningful hierarchical maps can be obtained. Advantages include its robust handling of the complexities of vision, its appearance-based nature which does not require extensive estimation of metric information, and the possibility for efficient path planning [3]. Our method currently requires one to specify only the number of clusters. In future work we would like to automatically select the number of clusters as well. Online versions of the algorithm (see [4])are also interesting for further research.

References

[1] A.J.Davison and D.W.Murray. Mobile robot localization using active vision. *In Proceedings of the European Conference on Computer Vision*, 1998.

[2] P. Althaus, H. Ishiguro, T. Kanda, T. Miyashita, and H. I. Christensen. Navigation for human-robot interaction tasks. In *In Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1894–1900, April 2004.

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

[4] O. Booij, Z. Zivkovic, and B. Krose. Pruning the image set for appearance based robot localization. In *In Proceedings of the Annual Conference of the Advanced School for Computing and Imaging*, 2005.

[5] H. Choset and K. Nagatani. Topological simultaneous localisation and mapping: Towards exact localisation without explicit localisation. *IEEE Transactions on Robotics and Automation*, 17(2):125–137, April 2001.

[6] E.Remolina and B.Kuipers. Towards a general theory of topological maps. *Artificial Intelligence*, 152:47–104, 2004.

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

[8] S. D. Jones and J. L. Crowley. Appearance based processes for visual navigation. *IEEE International Conference on Intelligent Robots and Systems, France*, 1997.

[9] J.Shi and J.Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Anlysis and Machine Intelligence*, 22(8):888–904, 2000.

[10] J. Kosecka and F. Li. Vision based markov localization. *In Proceedings of the IEEE Robotics and Automation conference*, 2004.

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

[12] Benjamin Kuipers, Joseph Modayil, Patrick Beeson, Matt MacMahon, and Francesco Savelli. Local metrical and global topological maps in the hybrid spatial semantic hierarchy. In *In Proceedings of the International Conference on Robotics and Automation ICRA*, New Orleans, May 2004.

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

[14] S. Nayar, S. Nene, and H. Murase. Subspace methods for robot vision. *CUCS-06-95, Technical Report, Department of Computer Science, Columbia University*, 1995.

[15] P.Beeson, N.K.Jong, and B.Kupiers. Towards autonomous place detection using the extended voronoi graph. *In Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.

[16] P.Lamon, A.Tapus, E.Glauser, N.Tomatis, and R.Siegwart. Environmental modeling with fingerprint sequences for topological global localization. *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, USA*, 2003.

[17] I.Shimshoni R.Basri, E.Rivlin. Visual homing: Surfing on the epipoles. *International Journal of Computer Vision*, 33(2):117–137, 1999.

[18] R.Sim and G.Dudek. Learning and evaluating visual features for pose estimation. *Transactions on Robotics and Automation International Conference Computer Vision*, 1999.

[19] P. Sala, R. Sim, A. Shokoufandeh, and S. Dickinson. Landmark selection for vision-based navigation. *In Proceedings of the International Conference on Intelligent Robots and Systems, Japan*, 2004.

[20] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets or 'how do i organize my holiday snaps'. *In Proceedings of the European Conference Computer Vision*, 2002.

[21] F. Schaffalitzky and A. Zisserman. Automated location matching in movies. *Computer Vision and Image Understanding*, 92:236–264, 2003.

[22] S. Se, D.G.Lowe, and J.Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 8(21):735–758, 2002.

[23] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

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