Topological mapping, localization and navigation using image collections

Friedrich Fraundorfer[1][2]
[1]Institute for Computational Science
Swiss Federal Institute of Technology
Zurich (ETH)
CH-8092, Zurich, Switzerland
fraundorfer@inf.ethz.ch

Christopher Engels[2]
[2]Center for Visualization
and Virtual Environments
University of Kentucky
Lexington, KY, USA
engels@vis.uky.edu

David Nistér[2][3]
[3]Microsoft Live Labs
Microsoft Research
Redmond, WA, USA
dnister@microsoft.com

***Abstract*— In this paper we present a highly scalable vision-based localization and mapping method using image collections. A topological world representation is created online during robot exploration by adding images to a database and maintaining a link graph. An efficient image matching scheme allows real-time mapping and global localization. The compact image representation allows us to create image collections containing millions of images, which enables mapping of very large environments. A path planning method using graph search is proposed and local geometric information is used to navigate in the topological map. Experiments show the good performance of the image matching for global localization and demonstrate path planning and navigation.**

I. Introduction

Recently, [1] introduced a highly scalable and efficient image search scheme with real-time performance. Image retrieval can be performed in linear time with a very small time constant, and adding a new image to the database can be done in constant time. The compact representation of an image in the database can handle millions of images, with up to 1 million images at real-time frame rates. Based on this method we investigate a novel vision-based approach to topological mapping, localization and navigation with a single perspective camera. We represent the robot's world environment as a linked collection of way-point images. The collection is built online from camera images while the robot is exploring the environment. Links are created between sequential images and are inserted by image matching. A camera view that matches an image in the database is not added again but instead the view already in the database is re-used and a link is created. Loop closing is achieved inherently in the process. Global localization can be achieved at real-time frame rates. The image representation in addition allows us to compute the relative orientation to the image in the database which adds local geometric information to the topological approach. The robot can navigate in the environment by following a sequence of way-point images from the database. The directional information from the localization is used to guide the robot along the way-points. Path planning is done by graph search within the linked image collection. The robot may move along the temporally-linked images and use shortcut links created by the image matching. If the robot looses track of the way-point sequence it can start a random walk until it gets matching images again by global localization. Then a path to the goal from a new position can be computed. The topological approach is highly scalable, thereby allowing operation in large environments. This topological approach enables the robot to perform a variety of different tasks, such as exploration, homing and search-and-retrieve missions. In an exploration mission, the robot gathers information about

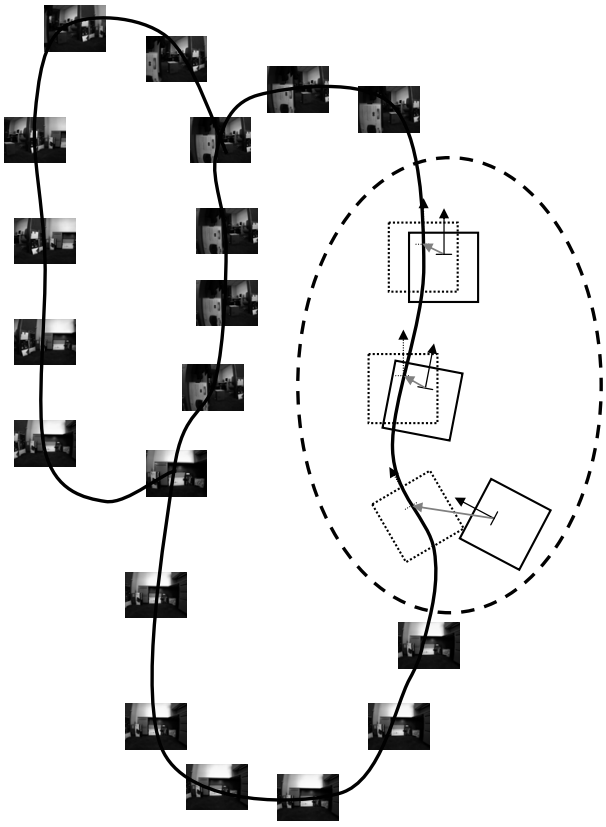

Fig. 1. The world environment is represented as a linked collection of way-point images. Image matching closes loops in the topological representation. Local geometric geometry allows a robot to follow a previously traversed path.

the topological structure of the environment by exploring without a goal. The created image collection can easily be used by other robots afterward. For homing, the robot could traverse back its route and the start position can be identified by global localization. In a search-and-return mission, a goal is set by presenting the image of a target location or target object. Target recognition operates in the same manner as localization: The robot performs a random walk or traverses previously explored paths from the image collection and returns if the object is found.

The paper is organized as follows: Section II will review related work and the current state-of-the-art approaches. Next, section III will discuss the image search scheme for global localization. Then, section IV, section V and section VI will discuss our new approaches to mapping, localization and navigation, respectively. Experimental results for localization, navigation and loop closing will be presented in section VII. Finally, section VIII concludes the paper.

II. Related work and state-of-the-art

Our method is related to the work of Goedeme *et al.* [2], [3]. They presented a system that follows a pre-executed visual path. The visual path is defined as a sparse sequence of omnidirectional images. To re-execute a visual path wide-baseline feature matching is used to match the current view to way-point images and to compute the directions from one way-point image to the next. SIFT-features [4] and invariant column segments [5] are used as local features. In addition to the directions from wide-baseline feature matching, information from frame-by-frame feature tracking is used to navigate from one way-point to the next. Map building is done offline and requires a training phase, unlike our approach where map building is done online. In addition their paper does not describe methods for path planning within the topological map. It also differs from our approach as the distance between way-points is about 2 to 4m, while our distances are much shorter, which increases the robustness of the wide-baseline matching. Their navigation approach is also limited to movements in a plane. More recent work focuses on dealing with self-similar environments [6]. Topological maps have been presented by a number of other authors. Zivkovic *et al.* [7] uses a topological world representation and wide-baseline feature matching. However, their work focuses on the creation of a hierarchical topological world representation. A topological map is also presented in Ulrich and Nourbakhsh [8]. Color histograms are used to perform place recognition but navigation has not been attempted. In an earlier work by Jones *et al.* [9], path following is performed by a global appearance-based method. However, matching was only achieved locally between the previous and next way-point: global localization is not possible. Visual navigation by path following was also presented by Royer *et al.* [10]. Their approach creates a world map in an offline process. First, image data is gathered by manually driving the robot through the environment. Features are tracked in the images and a 3D reconstruction of the features is computed. When re-executing the path the robot matches it's current camera view to the stored features and computes it's current pose from 3D-2D correspondences. The limits of this approach depend largely on the used features for tracking (Harris-Corners), the feature descriptor (normalized cross correlation) and the method for image matching (simple image by image matching). Using object recognition for loop closing has also been used sucessfully by Newman *et al.* [11], which uses the object recognition system described in [12]. Mapping is done with a 3D laser range finder while image matching is used to identify loops. The loops are geometrically closed by using both image and laser data.

III. Review of the image search scheme

The image search method by Nistér and Stewénius [1], also based on [12], plays an important role in our approach. Because of its compact image representation and its efficient matching process, it is well suited for mobile robot applications, including those operating in very large environments. The image search finds similar images by matching local features. First, the local features are detected in each image using the MSER detector [13]. Then a feature vector is computed over a local region using the SIFT descriptor [4]. Each SIFT feature vector is quantized into a vocabulary tree. A single integer value, called a visual word (VW), is assigned to a 128-dimensional SIFT feature vector. This results in a very compact image representation, where each image is represented by a set of visual words. Matching two images can be done by comparing the two lists of visual words. For image search applications the image database is set up as an inverted file. For each VW the inverted file maintains a list of image indices in which the visual word occurred. For an image query the indexed lists of all VW's that occur in the query image are processed. Weighted votes are collected for the images in the lists. The database image with the highest score is then selected as the best match. The query process is very efficient and fast. Tests show that a query in a 1 million image database (with an average number of 200 VW's per image) takes 0.02s. This results in a frame-rate of about 50Hz, well suited for real-time applications. Because of the compact image representation, a 1 million image database can be stored in less than 4GB, allowing it to be kept in RAM on current computers. Adding a new image to the inverted file database uses constant time only, since it is only necessary to add the image index to the according VW lists. This time is almost negligible compared to the query. MSER detection and SIFT feature computation can be done at a frame-rate of 15Hz on 640×480 images. Adding images at a frame-rate of 15Hz a 1 million image database will allow up to 18.5h of mapping operation. In [1] the authors reported that an image query on a 100 million image database took 6s with the database stored on hard disks and querying required to read the data from the hard disk. Although global localization can't be performed at frame rate anymore it could be done when required, e.g. after start up or to deal with the kidnapped robot problem.

IV. Topological mapping

The topological map consists of a collection of images (represented as VW's and stored as inverted file) and a link graph (represented as adjacency list). For each frame captured by the camera the VW representation is computed. Next, the database is searched for matching images. The inverted file lookup retrieves the top n-closest images in feature space. A geometric verification is performed that compares the spatial alignment of the feature correspondences within the query image and the database image. This is done by counting inliers in a robust homography estimation from the point matches[1]. With a RANSAC-scheme homography estimates are computed from 4-point sub-samples. We are not interested in the actual homography but we count the number of point correspondences that coarsely fulfill the homography. Depth variations are taken into account by using very loose thresholds when deciding between inliers and outliers. The geometric score is defined as the number of point correspondences that satisfy the homography constraint. A threshold on the geometry score is used to decide if the query image matches to an image in the database or if there is no match. If the image is not in the database the image will be added. Additionally a link between the new image and the previous image will be added to the link graph. In the case of finding a match in the database, the image is not added, only the link is created. This creates a loop in the topological structure. The map will be created online during exploration. In addition to the VW image representation we also store the image coordinates of the detected features. The image coordinates are used to compute the geometric score and to compute local geometric information during navigation. This results in an inverted file size of

$$DB_{inv} = 4fI, \tag{1}$$

where f is the maximum number of visual words per image and I is the number of images in the database. The factor 4 comes from the use of 4 byte integers to hold the image index where a visual word occurred. In addition the size for the additional geometry data computes as

$$DB_{geom} = 12fI. \tag{2}$$

For each detection the visual word and the image coordinates of the detection will be stored. Using a 4 byte integer for the VW and float data-type for the image coordinates this results in 12 bytes per detection.

V. Localization

Our scheme features global localization at frame rate. By image matching with the database the closest location within the topological map is computed. In addition to this topological place recognition the relative position to the database image is computed. The feature point correspondences are used to compute the relative orientation by using the 5-point algorithm [14]. The 5-point algorithm computes the essential matrix E, which encodes the rotation and translation of the cameras between two views. The robot's position is fixed to the camera position, so we assume that camera and robot coordinate systems coincide. Further, we assume a local robot-centered calibrated coordinate system with the camera position P_0 at the origin. P_1 represents the camera position and rotation of the matched database image. P_1 can be computed from the essential matrix and consists of a rotation R and a translation t where $P_1 = [R|t]$. R is a 3×3 rotation matrix and t is a 3×1 translation vector. The directional vector from the current image to the database image is given by

$$C = -R^T t. \tag{3}$$

Fig. 2 illustrates the relations. The rotation matrix R encodes the rotation of the robot after the robot has moved along C. We do not recover metric scale but use the directional information only for navigation.

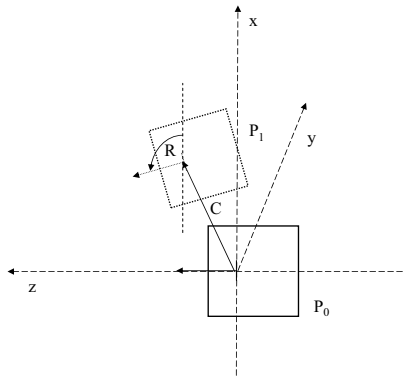

Fig. 2. Local geometry for the current view P_0 and a database match P_1.

VI. Navigation and path planning

Navigation in the proposed scheme works by traversing a sequence of way-point images. The way-point sequence is generated by graph-search based path planning. For path planning we show an image of the target location to the robot. The image is matched to the database and a path from the current location to the goal location is computed. Each edge in the graph has the same weight and the graph search will find the shortest sequence to the goal location. The robot can only move along previous routes, however the graph search may use shortcuts created from links inserted by image matching. For following a sequence of way-point images the directional information from the relative orientation estimation is used. We compute R and t from the current image to the first way-point image. The robot then moves into the direction C and rotates according to R to come close to the position where the first sequence image was taken from. Then the next way-point in the sequence is matched to the current image. Again R and t are computed and the robot moves on to the next way-point.

[1]However we do not assume a planar scene geometry.

A. Robot movements

The robot movements towards a way-point will consist of a series of fixed distance movements. Each straight movement into direction C is secluded by a turn specified by R to let the camera look into the target direction. After each such movement the direction to the way-point is re-estimated and adapted if necessary. A way-point is considered to be reached if the current image already matches to the image of the next way-point of the sequence. To be robust to inaccurate or incorrect relative orientation estimates, we limit the robot's maximum directional change and turn angle. We allow a maximal directional change of 10° in the direction of the way-point and a maximal turn angle of 5°. With these limits the target way-point is still in view of the camera even if the estimate was wrong and the robot moved into the opposite direction. Occasional bad estimates can be alleviated by this method. Fig. 3 illustrates the robots movements between way-points w_i and w_{i+1}. Being located at w_i the robot first computes the direction C_0 to w_{i+1} and the rotation R_0. It then performs a fixed distance movement into the direction of C_0 and performs a limited rotation according to R_0. This first step got the robot to location s_0. Due to inaccuracies in the movement s_0 might not precisely lie on the direction connection between w_i and w_{i+1} (as illustrated in the figure). Therefore the directions C_1, R_1 to w_{i+1} will be recomputed for the next step and after performing the movements the robot is located at s_1. These steps will be repeated until the goal position w_{i+1} is reached. In our illustration, the goal position gets reached after 4 steps at s_3. The goal position is not met precisely, as the robot moves a fixed distance every step. The robots navigational precision is limited by the fixed moving distance. With a minimum distance d a way-point can be reached with a precision of $\pm d$. However, these navigation errors don't accumulate as global localization is performed after each movement. Whether the goal position gets reached is determined by image matching. If the current view already matches to the next way-point image w_{i+2} the goal position is considered to be reached.

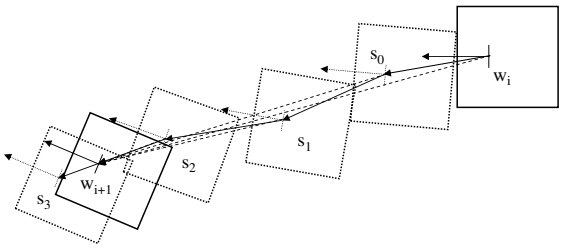

Fig. 3. Robot movement steps to move from way-point w_i to w_{i+1}.

VII. Experiments

Our experimental platform is a Pioneer DX. The robot is equipped with sonar sensors and a digital camera. The sonar sensors are used for collision detection only. The digital camera captures 640×480 pixel images with a frame rate of 15 Hz. A wide-angle lens with a large field of view is attached to the camera. The camera is mounted with an angle of 20° to the robots driving direction. The camera is calibrated and radial lens distortion is removed. Our navigation scheme requires that the camera is pointed towards a way-point at any time. For movements therefore our robot first needs to rotate into the driving direction, drive, and then rotate back so that the camera points to the way-point again. With a holonomic robot the navigation could be done without the extra rotations. It should be noted that our approach is not limited to robots equipped with a perspective camera. It can be adapted to work with omnidirectional images as well.

A. Localization experiment

To test the performance of the global localization we captured image sequences from two subsequent runs through a corridor (Corridor sequence). Each run constitutes a closed loop. The first run consists of 533 images, while the second run consists of 551 images. A database is created from the images of the first run. Each image of the second run is matched with the database. Fig. 4 shows the match similarity matrix. For each image from the second loop the best matching image from the first loop is marked in the matrix by a dot. The single consecutive trail shows that images from the second loop match with spatially close images from the first loop. The single dot in the bottom left corner is caused by a match between one of the first images of the second loop to one of the last images of the first loop because of an overlap in the image sequences. In addition we did a visual inspection of the matches and determined that for each image of the second run a correct corresponding image from the first run had been found. For each image match the geometric score (number of geometric consistent point matches) has been computed. Fig. 5 shows the histogram of the geometric score on the dataset. The lowest achieved geometric score was 9 point matches. It still represents a high confidence in the match but it might lead to inaccurate direction estimation for navigation. The histogram shows that most matches however are better conditioned. Fig. 6 shows two example matches.

B. Path following and homing

Tests of the visual path following and homing were carried out in an office environment. Fig. 8 shows the odometry data for the path following experiment overlayed with a sketch of the office environment. We drove the robot manually along a path from w_0 to w_n, with an approximate path length of 4m. A total of 118 images were captured during the run. Each third frame was stored in the database to build the topological map. The black curve is the manually driven path. The blue curve is the trajectory of the robot when re-executing the path

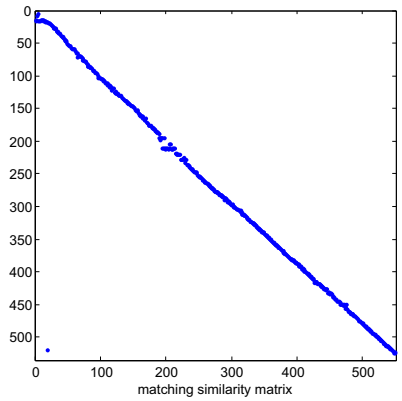

Fig. 4. Similarity matrix for matching the two corridor loops. For each image from the second loop the best matching image from the first loop is marked in the matrix by a dot. The diagonal structure shows that images from the second loop match with spatially close images from the first loop. The single dot in the bottom left corner is caused by a match between one of the first images of the second loop to one of the last images of the first loop because of an overlap in the image sequences.

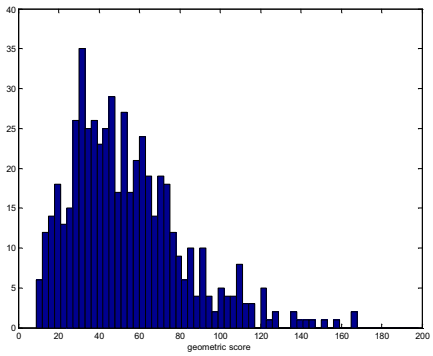

Fig. 5. Geometric score histogram for image matches from Corridor sequence. The lowest achieved geometric score was 9 point matches, which might lead to inaccurate direction estimation for navigation. However, most matches are better conditioned.

Fig. 6. Matching examples from the two loops of the Corridor sequence. Matching works well even for quite featureless images (a).

from start position w_0. The robot arrived at the goal position quite accurately. Fig. 7 shows a picture of the robot and the office environment. The results of the homing experiment

Fig. 7. The robot in the office environment.

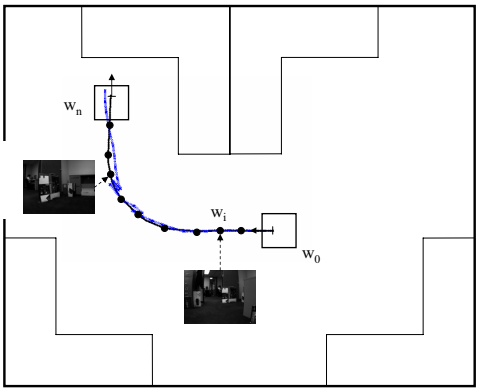

Fig. 8. Robot odometry for the path following experiment. The original path is in black. The robot successfully re-executed the path from w_0 to w_n by visual navigation (blue trajectory).

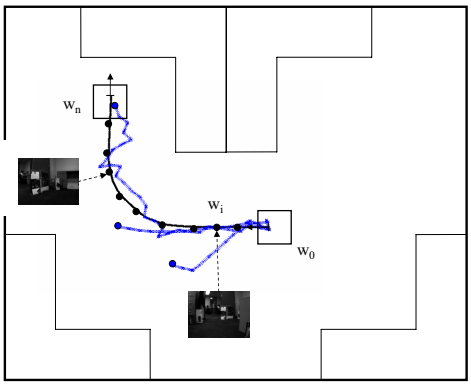

Fig. 9. Robot odometry for the homing experiment. The original path is in black. The robot successfully traversed the reversed path back to the start position w_0 from different start positions (blue trajectories).

are depicted in fig. 9. The robot odometry is shown for reversing the path from w_n to w_0 and two other different start positions. For homing the robot was driving backwards, the camera still pointing into the original direction and not in the new driving direction.

C. Loop closing

To test the loop closing capability the robot was put into random walk to explore the office environment. The robot

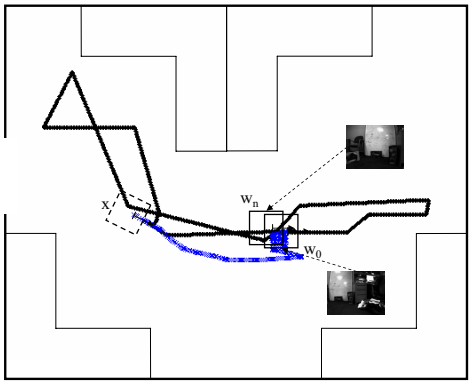

Fig. 10. Setup for the loop closing experiment. The robot randomly explored the room starting at w_0. A closed loop has been detected at w_n matching to the start point. Then the robot has been moved to position X and path planning to return home to w_0 has been initiated. The robot successfully performed global localization and computed a path using the shortcut provided by the closed loop. The robot then successfully executed the path and return to its home position. The traversed path is shown in blue.

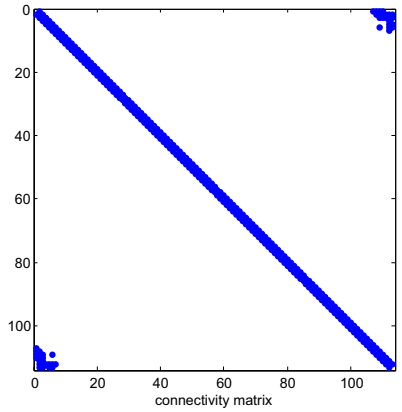

Fig. 11. Connectivity matrix for the loop closing experiment. The temporal connections of adjacent way-points form the diagonal matrix entries. The entries in the bottom left corner represent the closed loop between w_0 and w_n.

was instructed to stop when it matches its start position, i.e. it closes the loop. Fig. 10 shows the robot odometry. The start position is w_0. The robot headed out to the right. After a 180° turn it passed the start position, the camera facing in the opposite direction, thus unable to recognize it. After a turn on the other side it passed the start position again, the camera facing in the right direction and now it closed the loop at way-point w_n, like expected. The map so far contained 113 way-points. Next, we placed the robot at a new position X. It was instructed to perform self localization and compute a path to the start position w_0. The computed path consisting of 9 way-points uses the shortcut created by the closed loop between w_n and w_0, instead of simply reversing the whole forward path. The robot then successfully followed the path to the origin (blue trajectory). Fig. 11 shows the connectivity matrix of the topological map. The temporal connections of adjacent way-points form the diagonal matrix entries. The entries in the bottom left corner represent the closed loop between w_0 and w_n.

VIII. Conclusion

We presented a vision-based mapping, localization and navigation scheme that is scalable to very large environments. Based on an efficient image matching scheme [1] localization and navigation tasks can be performed in real-time. Even global localization is performed in real-time. A topological world representation consisting of an image collection with additional local geometry allows navigation without metric information. Loop closing is efficiently solved by the image matching scheme as well. The approach has been implemented on a Pioneer DX platform. Equipped with a single perspective camera the robot can drive around autonomously and create its world representation. We demonstrate the homing capabilities of the robot as well as its capability of re-executing paths. For homing the robot is able to use closed loops as short-cuts to get to the goal faster.

References

[1] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition, New York City, New York*, 2006, pp. 2161–2168.

[2] T. Goedemé, T. Tuytelaars, G. Vanacker, M. Nuttin, and L. Van Gool, "Feature based omnidirectional sparse visual path following," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton*, 2005, pp. 1003–1008.

[3] T. Goedemé, M. Nuttin, T. Tuytelaars, and L. Van Gool, "Markerless computer vision based localization using automatically generated topological maps," in *European Navigation Conference GNSS, Rotterdam*, 2004.

[4] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th International Conference on Computer Vision, Kerkyra, Greece*, 1999, pp. 1150–1157.

[5] T. Goedemé, T. Tuytelaars, and L. Van Gool, "Fast wide baseline matching for visual navigation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC*, 2004, pp. I: 24–29.

[6] ——, "Visual topological map building in self-similar environments." in *Proceedings of the Third International Conference on Informatics in Control, Automation and Robotics, Robotics and Automation, Setúbal, Portugal, August 1-5, 2006*, 2006, pp. 3–9.

[7] Z. Zivkovic, B. Bakker, and B. Kröse, "Hierarchical map building using visual landmarks and geometric constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton*, 2005, pp. 7–12.

[8] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," in *Proc. IEEE International Conference on Robotics and Automation*, April 2000, pp. 1023–1029.

[9] S. D. Jones, C. Andresen, and J. L. Crowley, "Appearance based processes for visual navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Grenoble*, 1997, pp. 551–557.

[10] E. Royer, M. Lhuillier, M. Dhome, and T. Chateau, "Towards an alternative gps sensor in dense urban environment from visual memory," in *Proc. 14th British Machine Vision Conference, London, UK*, 2004.

[11] P. Newman, D. Cole, and K. Ho, "Outdoor slam using visual appearance and laser ranging," in *IEEE International Conference on Robotics and Automation*, 2006, pp. 1180–1187.

[12] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE International Conference on Computer Vision, Nice, France*, 2003, pp. 1470–1477.

[13] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proc. 13th British Machine Vision Conference, Cardiff, UK*, 2002, pp. 384–393.

[14] D. Nistér, "An efficient solution to the five-point relative pose problem," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin*, 2003, pp. II: 195–202.