Actionable Topological Mapping for Navigation Using Nearby Objects

Junyoung Kim, Juyong Kim, Seungil You, Yoonseon Oh, and Songhwai Oh

Abstract— **In this paper, we propose a mapping and navigation method for mobile robots with low computational resources and limited memory capacity. The proposed navigation method is based on topological mapping of visual features for its compact representation and robustness in localization. In order to improve the localization accuracy and minimize the memory requirement, we propose the use of visual features from nearby objects. This paper presents a new topological map, called an *actionable topological map*, which is constructed using visual features from nearby objects and motion information between places in the map. By incorporating motion information, we make it easier for a robot to navigate using the compact representation of a topological map. The proposed method is suitable for light-weight, low-cost robotic platforms due to its low computational and space requirements. We demonstrate the effectiveness of our approach in experiments using an inexpensive, off-the-shelf robotic platform.**

I. Introduction

Localization and mapping are the most fundamental problems in robotics and solutions to these problems are required for many practical robotic applications [1]. When a robot enters an unknown place, it is required to localize itself while mapping its surroundings to perform its tasks. The mathematical formulation of this problem is known as simultaneous localization and mapping (SLAM) [2]. Various algorithms for solving the SLAM problem have been proposed and a number of different sensors have been used, such as laser, sonar, and visual sensors [2], [3].

Until recently, researchers in robotics have focused on exact mapping of surroundings [4], [5]. However, there are applications in which an exact map is not required. For example, if we are developing an indoor service robot, it may be enough to know which room (or which corner of the room) the robot is located in order to perform its task as long as the robot can avoid obstacles and move from one room to another. For this type of applications, if a robot can operate using a simpler map which requires less memory and computation, then we can implement light-weight localization and mapping algorithm on light-weight, low-cost robot platforms. While a simpler map is desired, the map needs to contain essential information to distinguish one place from another and a means for a robot to move from one place to another. For this objective, we propose an *actionable topological map* (ATM) using visual features, which is an augmented topological map with motion information. Furthermore, in order to improve the localization accuracy and minimize the memory requirement, our actionable topological map is built using visual features from nearby objects.

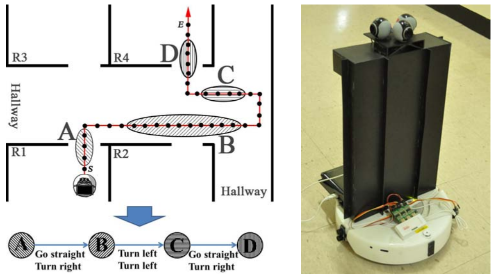
The best example of a topological map is the map of subway routes and stops. A topological map contains only vital information about the place without scale, distance, or direction information. However, the relationship between locations can be preserved in a topological map. In robotics, topological mapping is often constructed using visual words from images [6]–[11]. Visual words are representative visual features, such as scale invariant feature transform (SIFT) [12], and it has been reported many times that visual feature based localization approaches are robust against the challenging loop-closure problem and have received much attention recently [13], [14]. In [15]–[17], a topological map is used to solve a localization problem. But the required image processing step of their approaches requires heavy computation. In addition, the method for estimating a heading direction in [16], [17] requires an accurate localization on a topological map to navigate.

The proposed mapping and navigation method is inspired by how a human gives a direction. When someone gives you a direction to the place you have never been to before, she will explain the direction using noticeable landmarks, such as stores, restaurants, and buildings, and how you should navigate between those landmarks. Her direction will not contain detail information such as accurate distances between landmarks and angles of your heading at each instance. Our goal is to provide this type of brief instructions to a robot and make the robot to navigate its surroundings. In addition, we propose the use of visual features from nearby objects, called *nearby object SIFT* (nSIFT) features, to improve the localization accuracy and minimize memory and computation time requirements. These nSIFT features resemble noticeable landmarks that people use when they give directions.

We have implemented the proposed mapping, localization, and navigation methods on a light-weight mobile robot platform, iRobot Create [18], which is equipped with webcams. We present the experiment results which demonstrate that a mobile robot can easily localize itself and navigate along various paths using the proposed actionable topological map. Even when the objects that were used to construct the map were removed, the robot was able to localize itself using the map, showing the robustness of the proposed method. While the mapping phase requires off-line computation, localization and navigation can be done comfortably in real-time, thanks to the compact representation of an actionable topological

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2010-0013354).

Junyoung Kim, Juyong Kim, Yoonseon Oh, and Songhwai Oh are with the School of Electrical Engineering and Computer Science and ASRI, Seoul National University, Seoul 151-744, Korea (emails: {junyoung.kim, juyong.kim, yoonseon.oh, songhwai.oh}@cpslab.snu.ac.kr). Seungil You is with the Department of Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91125, USA (email: syou@caltech.edu).



(a) Concept (b) Platform

Fig. 1. (a)An example of actionable topological mapping. A robot moves from location S to location E and takes images of surroundings. The trajectory of the robot is shown as a solid red line. The black dots on the solid line are the locations at which images are taken. See the text for more detail. (b) The mobile robot platform based on iRobot Create used in experiments.

map.

This paper is structured as follows. Section II gives an overview and describes actionable topological mapping (ATM). Section III describes a robust localization method based on Bayesian filtering. The navigation method using an ATM is described in Section IV. The experimental results are given in Section V.

II. Mapping

Suppose that a robot travels from location S to location E. The robot is equipped with a camera and takes images as it moves. The robot also records control commands as it executes. For each image, we extract nSIFT features (described below) and cluster them to form a collection of visual words. An actionable topological map (ATM) is constructed from the collection of visual words and the history of control commands. An ATM is a graph $G = (V, E)$, where the vertex set V is a collection of nodes and E is an edge set. The locations with distinctive visual words form nodes in an ATM. A node can contain more than one consecutive locations at which images were captured if they contain similar visual words. If $(u, v) \in E$, two nodes u and v are closely located in the physical world and a robot can move from u to v. A unique feature of an ATM is that, for each edge, a compound action is associated. If $(u, v) \in E$, a compound action C_{uv} is a collection of control commands used by the robot to move from node u to node v. Figure 1(a) shows an example (the top diagram shows a trajectory of a robot and the corresponding ATM is shown below). For this example, four nodes are detected (A, B, C, and D) and a compound action is associated with each edge. The robot stores command history and makes an edge from these commands. For instance, C_{CD} is a compound action required to move a robot from node C to node D and contains two control commands (go straight and turn right).

We will now describe each step of our mapping algorithm in detail.

A. nSIFT

Based on the idea that nearby objects can provide better localization, we extract visual features from nearby objects. By reducing the number of features extracted from images, we can also reduce both memory and computation time requirements. Features from nearby objects can be detected by using a stereo camera or taking two images at two different vantage points to compute disparity values of features. Taking the advantage of mobility of a robot, we take the latter approach. Therefore, we assume a robot moves a fixed distance each time.

A robot takes an image, moves forward by a fixed distance b, and takes another image. Here, we assume that the orientation of the camera is perpendicular to the motion of the robot, so that we can easily emulate a stereo camera. For each camera, Lowe's scale invariant feature transform (SIFT) features [12] are extracted. We then match the SIFT features from the first image to the SIFT features from the second image and compute disparity values of all matching features. Finally, nSIFT features are selected by the nearby object filter based on the computed disparity values.

An example of extraction of nSIFT features is shown in Figure 2. Figure 2(a) and 2(b) show SIFT features extracted from two images. Notice that the camera is moved to the right. Figure 2(c) shows matched features from two images. We filter out matched features if they have either very large (features from very near objects[1]) or small (features from far objects) disparity values. The selected nSIFT features are shown in Figure 2(d). These nSIFT features are used to construct an ATM.

To find matching SIFT descriptors, we use the *Hellinger distance* [19] since the Hellinger distance performs better than the usual Euclidean distance. The Hellinger distance is used to measure the distance between two l_1 normalized histograms. The Hellinger distance of two l_1 normalized histograms $\tilde{h}_t$ and $\tilde{h}_s$, e.g., SIFT descriptors, is computed as

$$d^2_{Hellinger} = 1 - \sum_{i=1}^{n} \sqrt{\tilde{h}_t(i)\tilde{h}_s(i)}. \qquad (1)$$

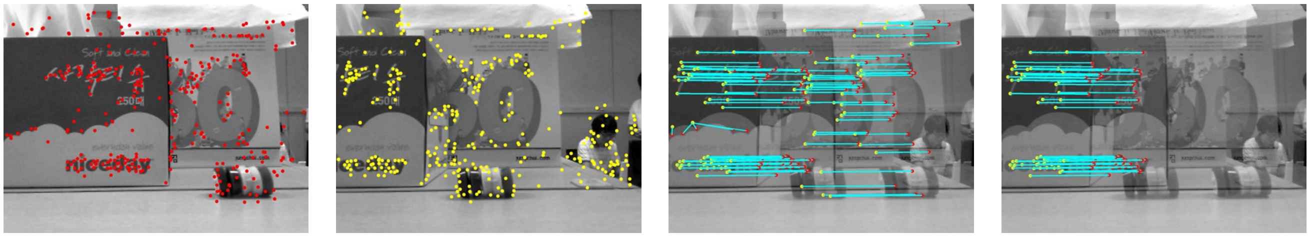
The Hellinger distance can vary from 0 to 1, 0 for identical distributions, and 1 when there is no overlap. The Hellinger distance is a metric.

B. Visual words

A *topological map* is a simplified map which shows only vital information and may not have detailed information. In visual mapping, a topological map is usually constructed from visual words, which are a collection of representative visual features, such as SIFT [7]. For each location at which an image is taken, a node in a topological map is assigned and each node contains visual words taken from the image. The relationship between a pair of nodes is described by similarity between distributions of visual words of two nodes.

We use a *hierarchical vocabulary tree* [20] to generate visual words from extracted nSIFT features. The hierarchical structure makes it possible to generate visual words with less computation time. Furthermore, the task of finding a visual

[1]We have found that features from very near objects are usually outliers as the robot moves at a certain distance away from the wall.



(a) (b) (c) (d)

Fig. 2. An example of extracting nSIFT features from two images. (a) An image taken at the previous location. The red dots represent positions of detected SIFT features. There are 398 SIFT features. (b) An image taken at the current location. The yellow dots represent positions of detected SIFT features. There are 423 SIFT features. (c) Matching features between two images (a) and (b). A total of 137 matches are found. (d) Filtered nSIFT features. There are 77 nSIFT feature pairs.

word for a given nSIFT feature is faster when a hierarchical vocabulary tree is used. A hierarchical vocabulary tree is constructed by running the k-means algorithm [21] recursively. For example, if we want to generate 10,000 visual words, it requires a tree with four levels with $k = 10$. Given a vocabulary tree, it requires at most 40 comparison operations before a corresponding visual word is found, instead of direct 10,000 comparison operations.

For better visual feature matches, we modified the k-means algorithm by replacing the Euclidean distance with the Hellinger distance.

C. Actionable topological mapping

Let n be the total number of visual words and I_t be the image taken at time t. Let $h_t^0(\cdot) : \{1, \ldots, n\} \rightarrow \mathbb{N}$ be a histogram of visual words from I_t and $h_t^0(i)$ is the number of occurrences of the i-th word from I_t. Let δ_t be a unique word indicator function from the histogram of visual words and it is defined as follows:

$$\delta_t(i) = \begin{cases} 1 & \text{if } h_t^0(i) > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

If a visual word appears in many images, this word is not effective for disambiguating locations. Hence, we borrow the idea from information retrieval [22], [23] and weight each word by its importance, i.e., the i-th word is weighted by $\log(N/n_i)$, where N is the number of nodes and n_i is the number of times the i-th word appears in the sequence of nodes. Based on these weights, we compute a weighted indicator vector $\hat{\delta}_t$ for image I_t, such that $\hat{\delta}_t(i) = \log(N/n_i)\delta_t(i)$ for all i.

The image I_t is declared as a node in an ATM if $\sum_{i=1}^{n} \hat{\delta}_t(i) > \theta_1$, where θ_1 is a threshold. If the previous image I_{t-1} is not a node in the ATM, I_t is declared as a new node. If I_{t-1} is a node, we measure similarity between I_{t-1} and I_t using the histogram intersection kernel [24]:

$$k(\hat{\delta}_{t-1}, \hat{\delta}_t) = \sum_{i=1}^{n} \min(\hat{\delta}_{t-1}(i), \hat{\delta}_t(i)). \quad (3)$$

If $k(\hat{\delta}_{t-1}, \hat{\delta}_t) > \theta_2$, where θ_2 is another threshold, and the control command between I_{t-1} and I_t is GS, we merge I_t into the node containing I_{t-1}. Otherwise, I_t is declared as a new node. By repeating this process for all consecutive pairs of images, we find nodes for the ATM. For robustness, a node is declared when three (or more) consecutive images satisfy the threshold θ_1. The threshold parameters are required to be chosen carefully to make the trade-off between the complexity of the ATM and the resolution of the map. When there are many nodes, the ATM can be complex but the resolution of localization is finer. On the other hand, when there are fewer nodes, the resolution of localization is coarse but we have a simpler ATM.

A pair of consecutive nodes in an ATM is connected by an edge. We associate each edge with a compound action which is a collection of all control commands between the nodes connected by the edge. For example, in Figure 1(a), $C_{AB} = \{GS, RT, GS, GS, GS, GS\}$, $C_{BC} = \{GS, GS, LT, GS, GS, GS, GS, LT, GS, GS\}$, and $C_{CD} = \{GS, GS, RT, GS, GS\}$.

III. Localization

In order to perform various tasks, a robot needs to localize itself with respect to the ATM. Since the robot's position and orientation can be different from its first visit or the map may have been built by another robot, a deterministic localization method does not provide reliable results. Hence, a probabilistic method is applied for localization using an ATM.

A. Bayesian filtering

Consider an ATM represented as a graph $G = (V, E)$. Suppose there are m nodes in the ATM and $V = \{v_1, \ldots, v_m\}$. Let $X_k \in V$ be the location of a robot at time k. With respect to the ATM, we can consider that a robot is moving along the directional edges of the ATM. Let Z_k be the measurements collected at time k. In our case, the measurement Z_k is the visual word indicator vector, i.e., $Z_k = \hat{\delta}_k$. Let $Z^k = \{Z_1, Z_2, \ldots, Z_k\}$, a collection of all past measurements up to time k. The goal of filtering is to compute $P(X_k|Z^k)$ for every k.

We denote the event $\{X_k = v_j\}$ by V_k^j. Applying Bayes' rule, we have

$$\begin{aligned} P(X_k|Z^k) &= \frac{P(Z_k|X_k, Z^{k-1})P(X_k|Z^{k-1})}{\sum_{j=1}^{m} P(Z_k|V_k^j, Z^{k-1})P(V_k^j|Z^{k-1})} \\ &= \frac{P(Z_k|X_k)P(X_k|Z^{k-1})}{\sum_{j=1}^{m} P(Z_k|V_k^j)P(V_k^j|Z^{k-1})}, \end{aligned}$$

where the second equality is due to the Markov property. $P(V_k^j|Z^{k-1})$ can be computed as follows:

$$\begin{aligned} P(V_k^j|Z^{k-1}) &= \sum_{r=1}^{m} P(V_k^j, V_{k-1}^r|Z^{k-1}) \\ &= \sum_{r=1}^{m} P(V_k^j|V_{k-1}^r)P(V_{k-1}^r|Z^{k-1}). \end{aligned}$$

Hence, the posterior $P(X_k|Z^k)$ can be computed recursively, if the dynamics model $P(X_k|X_{k-1})$, i.e., $P(V_k^j|V_{k-1}^r)$, and the measurement model $P(Z_k|X_k)$ are available.

B. Dynamics model

The dynamics model is $P(X_k|X_{k-1})$, which is the transition probability of moving from node X_{k-1} at time $k-1$ to X_k at time k. For simplicity, we assume that the robot follows a random walk on the ATM and use the following dynamics model:

$$P(X_k|X_{k-1}) \propto \exp\left(-\frac{1}{\sigma_{dist}^2} d_X(X_k, X_{k-1})\right), \quad (4)$$

where $d_X(u,v)$ is the distance between node u and node v, i.e., the length of the shortest path from v to u in the ATM. σ_{dist}^2 is a scaling parameter and defined as $\sigma_{dist}^2 = \kappa\|C_{X_{k-1}X_k}\|$, where κ is a constant and $\|C_{X_{k-1}X_k}\|$ is the number of action commands from X_{k-1} to X_k. Hence, σ_{dist}^2 is large for a longer distance and small for a shorter distance.

C. Measurement model

The measurement model $P(Z_k|X_k)$ is defined as follows:

$$P(Z_k|X_k) \propto \; s_Z(Z_k, Z(X_k)), \quad (5)$$

where $s_Z(x,y)$ is a similarity function, giving the similarity between x and y, and $Z(X_k)$ is the measurements associated with the node X_k. In our case, $Z(X_k)$ is the weighted indicator vector stored in the node represented by X_k of the ATM. We used the set intersection measure as the similarity function s_Z as follows:

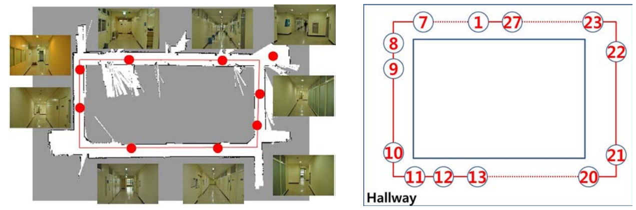
$$s_Z(\hat{\delta}_t(i), \hat{\delta}_s(i)) = \frac{\sum_{i=1}^{n} \min(\hat{\delta}_t(i), \hat{\delta}_s(i))}{\max(|\hat{\delta}_t|, |\hat{\delta}_s|)}, \quad (6)$$

where $|\hat{\delta}| = \sum_{i=1}^{n} \hat{\delta}(i)$.

IV. Navigation

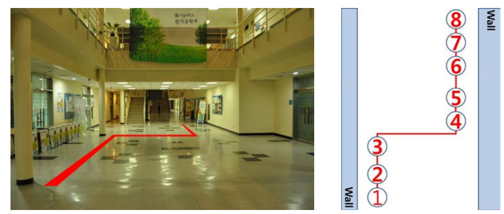
To navigate a robot from one node in an ATM to another node, we can first check if there is a valid path, a path connecting two nodes with directed edges in the graph. If a valid path is found, we need to determine the current heading of the robot. By computing two posterior distributions, one for the forward movement and another for backward, we can find the heading of the robot.

Next, we search for the shortest path and the robot moves along the shortest path. Consider the edge (u,v) of the shortest path. When a robot moves from node u to node v on the ATM, we simply apply the compound action C_{uv} to move the robot. However, due to the odometry error, the robot may have not arrived at v or it may have passed v. Hence, after executing the compound action, the robot localize itself using the localization method described above until a known node is found. If the robot detects node v, it moves forward by executing a series of GS control commands until the robot leaves the node.



(a) (b)

Fig. 3. (a) A map of the hallway used in the first experiment. The red line is the path of a mobile robot during the mapping phase. The length of the path is about $103\,m$. The red dots are locations where the images are captured. (b) An ATM constructed by the proposed algorithm.



(a) (b)

Fig. 4. (a) A photo of the lobby used in the second experiment. The red path indicates the trajectory of the mobile platform during the mapping phase. The length of the path is about $38\,m$. (b) An ATM constructed by the proposed algorithm.

V. Experimental results

The mobile robot platform used in the experiment consists of iRobot Create [18], a HP Mini 110 netbook, and two Logitech C250 webcams (see Figure 1(b)). Two webcams are placed perpendicular to the direction of motion of the mobile robot in order to emulate a stereo camera using a single camera (for each side).

The first experiment was conducted in a hallway as shown in Figure 3(a). Since the hallway does not have many distinctive visual features and similar patterns repeat, it is an extremely difficult environment for localization. The length of the path that the robot has traveled is about $103\,m$ and the path contains four left turns and one loop. The second experiment was conducted in a lobby as shown in Figure 4(a). In this experiment, we demonstrate the effectiveness of nSIFT for navigating in a wide area.

A. Actionable topological mapping

In the hallway, the robot took 1,029 images from each camera, a total of 2,058 images. The robot also recorded control commands between each image captures. Between each consecutive image captures, the robot was commanded to move at the speed of $10\,cm/s$ for one second, hence, the baseline distance $b = 10\,cm$ in our experiments. However, the actual length of each movement was between $10\,cm$ and $12\,cm$ due to the limitations of hardware. From 2,058 images, a total of 109,679 SIFT features were detected and, among these SIFT features, 14,122 nSIFT features were found. Hence, there was a reduction of 87.1% compared to regular SIFT features when nSIFT features are used. We selected features with disparity values between 30 pixels and 100 pixels as nSIFT features.

On the other hand, in the lobby, the robot took 381 images from each camera, a total of 762 images. The number of images of the second experiment was less than first experiment, but a total number of 411,488 SIFT features were extracted. 51,914 nSIFT features were found, a reduction of 87.4%.

We now describe the hallway experiment in detail. We constructed a hierarchical vocabulary tree with four levels and five branches to make 625 visual words. From our experiment, we found that 625 visual words were enough to disambiguate locations. We think that it was possible because nSIFT features are highly distinctive for the purpose of localization.

For the construction of an ATM, the following threshold parameters were used: $\theta_1 = 11.554$ and $\theta_2 = 25.0373$. θ_1 was set by computing the mean value of image scores (for an image I_t, its score is computed as $\sum_i^n \hat{\delta}_t(i)$). θ_2 is set by computing the mean value of different score values between $\hat{\delta}_{t-1}$ and $\hat{\delta}_t$. Figure 5 shows the score of nSIFT features for each left-right image pairs and detected nodes for the ATM. The algorithm detected 27 nodes. Note that the lengths between nodes are not proportional to the actual distances the robot traveled. Similarly, an ATM of the lobby was made in the same way. Figures 3(b) and 4(b) also show the resulting ATMs.

B. Localization

Once an ATM is constructed, a robot can be localized with respect to the ATM as explained in Section III.

We placed a robot between nodes and commanded the robot to move to a certain node. We assumed that the robot does not know its current location when it starts moving, i.e., it also needs to solve the kidnapping problem. We considered three cases: (Case 1) the robot moves in the hallway from node 10 to node 18; (Case 2) the robot moves in the lobby form node 8 to node 1; (Case 3) the robot moves the same as Case 2 but all SIFT features are used for localization. For Case 1 and Case 2, nSIFT features are used for localization.

Figure 6 shows the posterior of the current location computed by the Bayesian filter at different times for Case 1. While the hallway is featureless and difficult to localize using visual features, the proposed algorithm which combines nSIFT and a Bayesian filter correctly determines the current location of the robot.

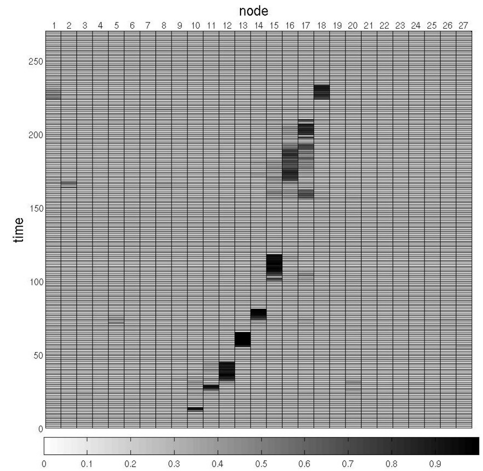
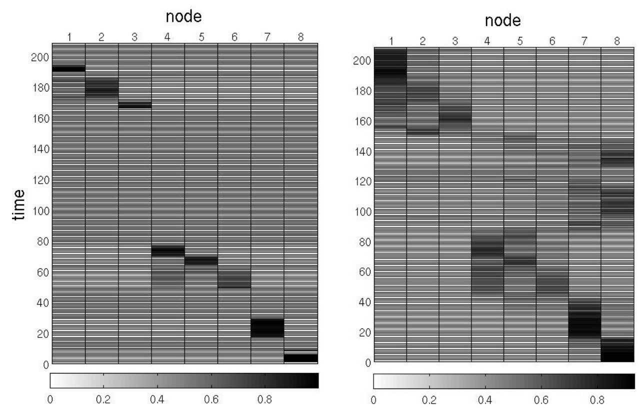


Fig. 6. The posterior of the current location at different times for Case 1. A robot moved from an edge (between node 9 and node 10) to node 18 in the hallway. When a robot started moving, the robot assumed a uniform prior on its starting location.



(a) Case 2 (b) Case 3

Fig. 7. Posteriors of the current location at different times for Case 2 (nSIFT features are used) and Case 3 (all SIFT features are used). A robot moves from node 8 to node 1 in the lobby. When a robot starts moving, the robot assumes a uniform prior on its starting location.

The localization results for Case 2 and 3 are shown in Figure 7. We can verify that nSIFT is better suited for localizing a robot (Case 2) than using all SIFT features (Case 3). Hence, we can use a less number of features for better localization using nSIFT, which is ideal for light-weight, low-cost robots.

C. Navigation

We have conducted a number of experiments using different starting and ending locations in both environments. First, a user specifies the destination node in the ATM and command the robot to move. Then the robot localizes itself on the ATM. When a robot is on a node of the ATM, it moves forward $10\,cm$. If a robot is at an edge of the ATM, the compound action associated with the edge is executed.

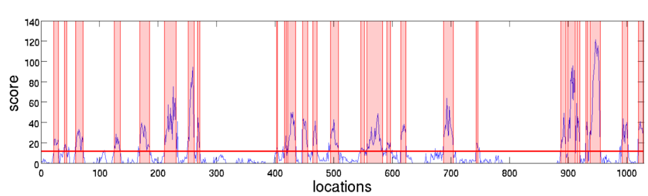


Fig. 5. Scores of nSIFT features for a sequence of 1029 left-right image pairs. The score for image I_t is computed as $\sum_{i=1}^n \hat{\delta}_t(i)$. Light red regions are declared as nodes. A bold red line shows the level of θ_1.

Environment	command	start	destination	trajectory
hallway	visit 11, node 9	edge 9-10	node 9	10-11-10-9
hallway	node 18	edge 9-10	node 18	10-11-…-18
hallway	node 9	edge 5-6	failed	6-fail
lobby	node 8	node 1	node 8	1-2-…-8
lobby	node 1	node 8	node 1	8-7-…-1
lobby	node 2	edge 3-4	node 2	4-5-4-3-2
lobby	node 2	node 8	failed	8-7-6-5-4-fail

TABLE I

ATM-BASED NAVIGATION RESULTS

When a robot starts in the hallway, the robot does not know its current location and its heading direction. The robot is placed on an edge between 9 and 10, we gave a command to the robot to visit node 11 and return to node 9. The robot moves forward and detect its location as node 10 and detects its heading direction. The robot executes $C_{10,11}$ to visit node 11. The compound action command $C_{10,11}$ contains a LT command and the robot makes a left turn successfully. When the robot arrives at node 11, it turns around and execute $C_{11,10}$ and $C_{10,9}$ to return to node 9. Notice that the compound action commands $C_{11,10}$ and $C_{10,9}$ are constructed on the fly. For example, the LT command in $C_{10,11}$ is replaced by a RT command in $C_{11,10}$. The localization was relatively easier in the lobby since each node contains a large number of distinctive visual words. The experimental results on navigation is summarized in Table I. The mobile robot successfully navigated using the ATM except two cases. In the first failed case, the edge between node 7 and node 8 of the ATM constructed for the hallway contains RT command right after node 7. The robot failed to detect node 7 and kept moving forward to search for a known node until it hit the wall. We plan to correct this problem by moving a robot in the hallway until it localizes while avoiding a front wall using a sonar sensor. The second failed case occurred in the lobby. The robot did not move straight and moved away from the desired trajectory and left the area for which the map was built. Again, we plan to correct this problem using sonar sensors so that the robot can move straight along the wall while keeping a distance from the wall.

VI. Conclusion

We presented an actionable topological map which includes compound actions such that a robot can navigate the physical world using a topological map. In order to improve localization accuracy and reduce memory and time requirements, we have introduced nSIFT features. In addition, we instrumented a mobile robot platform with two side cameras for better localization. The experimental results show that the proposed method is a promising approach for a light-weight robot platform to navigate in a complex environment when accurate metric localization is not required. We are currently developing a method to maintain a fixed distance from the wall and avoid obstacles using sonar sensors.

References

[1] S. Thrun, "Simultaneous localization and mapping," *Robotics and cognitive approaches to spatial mapping*, 2008.

[2] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT Press, September 2005.

[3] J. Aulinas, Y. Petillot, J. Salvi, and X. Lladó, "The SLAM problem: a survey," in *Proc. of the conference on Artificial Intelligence Research and Development: Proc. of the 11th International Conference of the Catalan Association for Artificial Intelligence*. IOS Press, 2008.

[4] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2001.

[5] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, pp. 3–20, 2006.

[6] O. Booij, B. Terwijn, Z. Zivkovic, and B. Krose, "Navigation using an appearance based topological map," in *Proc. of IEEE International Conference on Robotics and Automation*, 2007.

[7] F. Fraundorfer, C. Engels, and D. Nister, "Topological mapping, localization and navigation using image collections," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, November 2007, pp. 3872–3877.

[8] D. Kortenkamp and T. Weymouth, "Topological mapping for mobile robots using a combination of sonar and vision sensing," in *Proc. of the twelfth national conference on Artificial intelligence*. American Association for Artificial Intelligence, 1994.

[9] T. Goedemé, M. Nuttin, T. Tuytelaars, and L. V. Gool, "Omnidirectional vision based topological navigation," *International Journal of Computer Vision*, vol. 74, pp. 219–236, 2007.

[10] Y. Jiang, C. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *Proc. of the 6th ACM international conference on Image and video retrieval*. ACM, 2007.

[11] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, p. 647, 2008.

[12] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[13] A. Angeli, D. Filliat, S. Doncieux, and J. Meyer, "Fast and incremental method for loop-closure detection using bags of visual words," *IEEE Trans. on Robotics*, vol. 24, no. 5, pp. 1027–1037, October 2008.

[14] M. Cummins and P. Newman, "Probabilistic appearance based navigation and loop closing," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2007.

[15] O. Koch and S. Teller, "A self-calibrating, vision-based navigation assistant," *International Journal of Computer Vision Applications for the Visually Impaired*, 2008.

[16] J. Wolf, W. Burgard, and H. Burkhardt, "Robust vision-based localization by combining an image-retrieval system with monte carlo localization," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 208–216, 2005.

[17] O. Koch, M. Walter, A. Huang, and S. Teller, "Ground robot navigation using uncalibrated cameras," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2010.

[18] "iRobot." [Online]. Available: http://www.irobot.com/

[19] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.

[20] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[21] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

[22] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval*. ACM press New York., 1999, vol. 463.

[23] J. Sivic and A. A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. IEEE International Conference on Computer Vision*. IEEE Computer Society, 2003.

[24] M. Swain and D. Ballard, "Color indexing," *International journal of computer vision*, vol. 7, no. 1, pp. 11–32, 1991.