Scene Recognition with Omnidirectional Vision for Topological Map using Lightweight Adaptive Descriptors

Ming Liu, Davide Scaramuzza, Cédric Pradalier, Roland Siegwart and Qijun Chen

***Abstract*— Mobile robots rely on their ability of scene recognition to build a topological map of the environment and perform location-related tasks. In this paper, we describe a novel lightweight scene recognition method using an adaptive descriptor which is based on color features and geometric information for omnidirectional vision. Our method enables the robot to add nodes to a topological map automatically and solve the localization problem of mobile robot in realtime. The descriptor of a scene is extracted in the YUV color space and its dimension is adaptive depending on the segmentation result of the panoramic image. Furthermore, the descriptor is invariant to rotation and slight changes of illumination. The robustness of the scene matching and recognition is tested through real experiments in a dynamic indoor environment. The experiment is carried out on a mobile robot equipped with an omnidirectional camera. In our tests, the average processing time is 30 ms for each frame including feature extraction, matching, and the adding of new nodes.**

I. INTRODUCTION

In this paper we propose a lightweight descriptor for omnidirectional vision. It enables the mobile robot to recognize scenes based on image appearance and autonomously add nodes into a topological map while fitting the realtime requirement.

One classic application of navigation based on topological map is the robot homing scenario. The aim is to enable the robot to go to a predefined location only by knowing the target and the current image through vision [2][10]. Omnidirectional vision has shown to be one of the most suited sensors for this task because its 360° field of view [3], [11]. Another reason for choosing omnidirectional vision is that, when the camera is mounted perpendicularly to the plane of motion, the vertical lines of the scene are mapped into radial lines on the image. Therefore, a descriptor based on segmentation by vertical edges can be envisaged.

We employ the unwrapped image to simplify the extraction of descriptors in this work. Based on the unwrapped image, Booij et al. [1] built topological maps using SIFT features [5] from unwrapped panoramas. Menegatti et al. [6], [7] (for metric based map) had a robot to navigate in a one-room scene by using the Fourier transforms of the unwrapped images. A common drawback of these methods is that they omit the color information. In fact, color could provide more direct features without complicated computational cost. By analyzing the sequence of color blobs in the panoramic image, Lamon et al. [4] developed a descriptor for panoramic images called "fingerprint" of the environment. An efficient method for matching the fingerprints was also developed. Tapus et al. [12] applied the "fingerprint" approach of Lamon and built topological maps of multiroom indoor environments. However, both Lamon and Tapus used the measurements from a laser range finder to boost the matching of the descriptors. Conversely, in our work we introduce a descriptor extracted only by omnidirectional images and without the use of a range finder: the laser range finder is not suitable for housekeeping applications due to its high cost and the potential damage to people eyes.

Most of the state-of-art works use keypoints detection and matching [14][1][3], which has high computational cost as the point descriptors have usually high-dimension. This makes the mobile robot hardly work in real-time (e.g. 20 Hz) on other tasks, like motion planning or obstacle avoidance, besides recognizing places. A lightweight and efficient descriptor would help to preserve more time slots from the CPU for other tasks. Scaramuzza et al.[11] proposed a robust descriptor for tracking vertical lines in omnidirectional images in real-time (20 ms). Murillo et al.[8] also proposed a descriptor for vertical lines and a pyramidal matching method for metric localization, and used the number of lines in each image for topological localization. The recognition time was around 0.35s with 40 reference views according to their results.

Considering the characteristics of indoor environments, we hypothesize a basic fact: the important vertical edges naturally divide the indoor environment into several meaningful cuts. For example, the edges of windows, doors, cabinets, or bookshelves determine different areas with distinct characteristics. The color features in these different cuts are usually distinguishable because of the different texture of objects. Based on this fact, a new color based descriptor called FACT (Fast Adaptive Color Tags) is introduced in this work.

The objectives that we want to achieve with this paper are as follows:

- a robust appearance based segmentation method for unwrapped panoramas;
- an adaptive and robust descriptor for indoor environments;
- automatically recognizing and generating nodes for the topological map, while fitting the realtime requirement.

This paper is organized as follows. Section II describes our vertical edge based segmentation method. Section III explains how to extract the FACT color tags from the

This work was supported by CSC (China Scholarship Council) and Robots@home STREP Project IST-6-045350, and partially supported by The National high technology Research and Development Program of China (863 Program) 2009AA04Z213

Ming Liu is with CEIE of Tongji University and Autonomous Systems Laboratory in ETH Zurich. ming.liu@mavt.ethz.ch

segmented image, the matching algorithm, and the gener of the topological map. The evaluation and experiment be found in Section IV, where we also make a compa between the computational cost of our approach and th the keypoint-based approach.

II. Segmentation of the Panorama

Since this work proposes a descriptor based on features and segmentation on the panoramic image robustness of the method for detecting vertical edges i of the key problems to solve. The segmentation is mad extending the dominant vertical lines in the panoramic i as depicted in Figures 1, 2, and 3.

After unwrapping the raw panoramic image (Fig. 1 apply in sequence Sobel filtering (only along the x direc Otsu thresholding [9], and morphological operators to e the most dominant vertical lines. Fig. 2 shows the resu vertical extraction process. Note, only half of the unwra image is shown here because of the space limitation.

Fig. 1. A unwrapped result

Fig. 2. An output of vertical edges detection

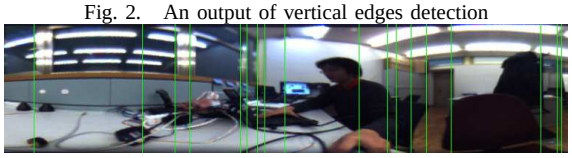


Fig. 3. The segmentation result

The dominant vertical lines are chosen based on their length. All the lines whose length is above average are retained. The morphological operators are used just to fuse those lines, which are too close to each other, into a single line.The detailed processing phases are shown in Fig. 4.

As observed in Fig. 3, the vertical lines partition the panoramic image into multiple regions. In the next section, we will explain how to extract our descriptor from these regions.

III. FACT Descriptor and Topological Mapping

In this section, we describe the method for building the FACT "*tags*" and its application to topological mapping.

In this work, we chose the YUV color space, where Y signal represents the overall brightness of the pixel and U-V are the two chromatic components. The benefit of this color space is that we only need two elements (i.e. U and V) to represent a color regardless of its brightness.

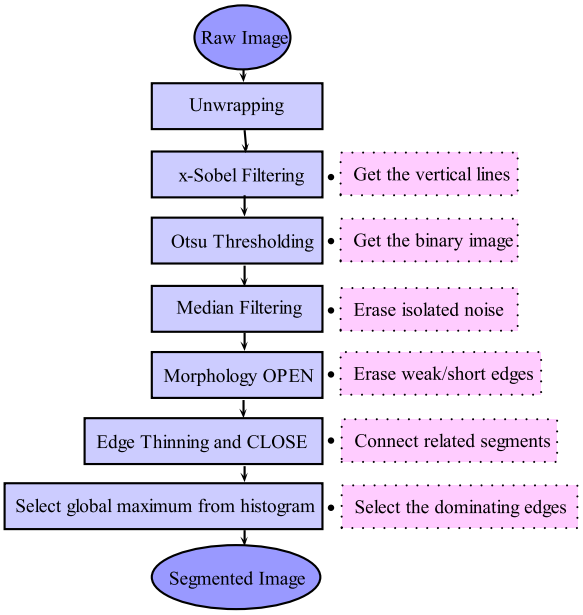


Fig. 4. The segmentation process

For each region between two vertical lines, the average color value in the U-V space is extracted. Comparing to other keypoint-based or edge-based descriptor, an obvious advantage in our approach is that the similarity between features in the U-V space will be simply measured in terms of a 2D Euclidean Distance.

A. Building the Descriptor

We extract the descriptor based on the segmented unwrapped image explained in the previous section. The descriptor is formed by the U-V color information and the width W (in pixels) of the region, which is delimited between two vertical edges. Instead of taking each pixel in every region into account, we directly use the average of U-V value that was calculated for each region. U_i and V_i indicate the color information of region i.

One primitive idea is that even if the width of each region may change during the translation of the camera, the projected area in the real-world can be well-determined in a local neighborhood, as long as the segmentation stays constant. In this case, the average value for a certain region in color space will keep constant. On the other hand, we must avoid the false positive caused by color similarity of regions. For example, the difference between a green cup and a green cabinet may be very small in color space, but the geometric features of these two are distinguishable. Therefore, we employ the width of correspond region W_i as the third dimension of our descriptor. By testing the ratio of the width of corresponding regions, the descriptor can get more reliable results. If we let N be the number of regions segmented from the unwrapped image[1], the dimension of the FACT descriptor of a scene is $3 \times N$. A sample descriptor D

[1]According to our experiment, N is usually smaller than 100 and greater than 20.

is shown in Eq. (1). Each column in the descriptor is named one *Tag*.

$$D = \begin{pmatrix} U_1 & U_2 & & U_N \\ V_1 & V_2 & \dots & V_N \\ W_1 & W_2 & & W_N \end{pmatrix} \quad (1)$$

B. Descriptor Matching

The matching stage is the fundamental part of our method. In this subsection, we will introduce how to evaluate the similarity between two descriptors. Let us assume that we have two descriptors D_1 and D_2, with dimension of $3 \times j$ and $3 \times k$ respectively. D_1 is a descriptor already stored in the database (e.g. from a previously visited location), and D_2 is a descriptor extracted from the current image. The purpose is to identify if the current location, where D_2 was taken, is the same (or identical) to the place where D_1 was extracted. Notice that j is usually different from k because in general the descriptors are extracted from different places.

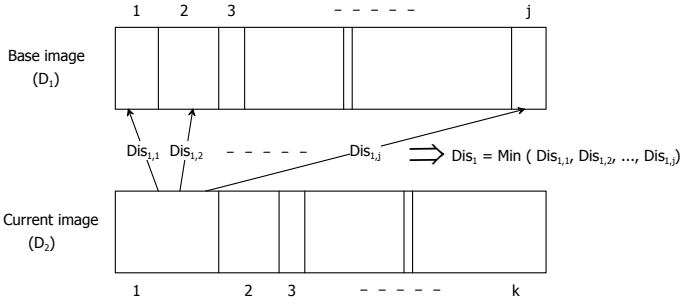


Fig. 5. A schematic diagram of Test 1 during the matching process.

In order to find the candidate match for a given descriptor, we devised the three tests listed below. Please notice that the angular order of the tags is also important in the matching stage. Here we just list the methods to compare the fundamental elements of FACT, such as color features and geometry.

Test 1: *Tag Matching in the U-V Color Space*
As shown in Fig. 5, the distance in the U-V color space is first calculated on each region in the current image using the 2D Euclidean Distance. The minimum of the distance array Dis_i is selected as represent of the region. If Dis_i is smaller than a very limited pre-defined threshold, TH_{local}[2], this region *i* is considered matching with the corresponding region in D_1 and passes the Test 1.

Test 2: *Tag matching in geometric space*
If the *Tag* passes the first test, the width of the region is then examined. Here the comparison is made in terms of the ratio between the widths of the two regions. For instance, if region 1 in the current image matches with region 4 of another image according to test 1, then the ratio between the width of region 1 in D_2 (namely $W_{2,1}$) and the width of region 4 in D_1 (namely $W_{1,4}$) must satisfy the inequality $\frac{1}{3} < \frac{W_{2,1}}{W_{1,4}} < 3$. The range of this ratio has been found empirically. If this test is passed, the corresponding region in the database will be eliminated when matching with other regions in the current image.

Test 3: *Descriptor matching*
Tests 1 and 2 are executed recursively until all the *Tags* in D_2 have been tested. The final score of current image is given by the ratio between the number of matching regions in the database and length of D_1, namely j.

It should be noticed that the FACT descriptor is a jointed feature of combined regions, and the influence of one single region is limited due to the big number of regions. Only the appearance of the entirety will determine the matching result. Therefore a trust-factor should be employed to compromise the importance of a single *Tag* and the entirety. When the dimension of the base descriptor is lower than the current descriptor, the multiplier $m = \frac{\#tags\ in\ current\ descriptor}{\#tags\ in\ base\ descriptor}$ is used as compensation, because the smaller number of regions may be caused by the occlusion of strong vertical edges. On the contrary, if the dimension of the base descriptor is higher (which means that the *Tags* in the current image are easier to match), $\frac{1}{m}$ is multiplied as a punishment. Assuming $\mathbb{M}$ represent the multiplier in the two cases, we get the resulting score of D_2 matching with D_1 as,

$$Score(D_2|D_1) = \frac{\#matched\ Tags\ of\ D_2}{\#total\ Tags\ of\ D_1}(\%) \cdot \mathbb{M}(\%) \quad (2)$$

C. Change detection and $Nodelist$

In this subsection, we will introduce how to use the FACT descriptor in the framework of topological map building and scene recognition. In practice, two reasons will affect the matching result. First, the segmentation result can be affected by noise or vibrations of the platform. For example, vibrations can make vertical lines non-radial anymore in the omnidirectional picture and so some weak edges may be missing. Secondly, dynamic objects will cause occlusions during the real-time experiment. For example, some of the moving objects or humans may occlude the existing strong edges. Hereby we set up two basic hypotheses according to these two reasons, respectively:

Hypothesis 1: each topological node will last at least for two frames. It will exclude the internal error caused by occlusions. As the camera will usually grab more than one frame per second, this is reasonable for this study.

Hypothesis 2: the dynamic objects will only cover no more than 30% of the area in the unwrapped image. As the omnidirectional camera covers 360°, we set a threshold that if more than 70% of the target image matches an existing node, then the current position will be considered matching

[2]We choose TH_{local} = 3e-4 in this work, and we used a 10×10 U-V color space. According to our test, this threshold applies very strict filtering of false positive.

$$\begin{cases} Nodelist = \{1, 2, 3...n\} \\ FACTlist = \left\langle \begin{pmatrix} U_{1,1} & U_{1,2} & & U_{1,j} \\ V_{1,1} & V_{1,2} & \cdots & V_{1,j} \\ W_{1,1} & W_{1,2} & & W_{1,j} \end{pmatrix}_1 \begin{pmatrix} U_{2,1} & U_{2,2} & & U_{2,k} \\ V_{2,1} & V_{2,2} & \cdots & V_{2,k} \\ W_{2,1} & W_{2,2} & & W_{2,k} \end{pmatrix}_2 \cdots \begin{pmatrix} U_{n,1} & U_{n,2} & & U_{n,l} \\ V_{n,1} & V_{n,2} & \cdots & V_{n,l} \\ W_{n,1} & W_{n,2} & & W_{n,l} \end{pmatrix}_n \right\rangle \end{cases}$$

Fig. 6. The pattern of $Nodelist$ and $FACTlist$

the database.

An incremental algorithm will automatically create nodes into a FACT-based list called $Nodelist$, and the descriptors of the nodes in the $Nodelist$ will be filled into $FACTlist$. The patterns of these two lists are shown in Fig. 6. By thresholding the scores of subsequent frames, i.e. if the matching resultsof two consecutive frames are lower than the predefined threshold TH_{global} (78% in this work), the current location is added as a new node into the topological map. Every time, the algorithm will compare the current descriptor with all the nodes in the $Nodelist$. Only the highest $Score$ of current image $Im(n)$ based on the $Nodelist$ is taken into account.

D. Refinement of Result

The incremental algorithm runs automatically. According to our test, a few redundant nodes may be created. The reason is that these nodes are mostly generated because some important vertical edges are occluded by obstacles or moving people when there are not many regions detected. We perform an off-line method for refining the $Nodelist$ in the reverse sequence as a supplement. Using this method, it costs less than 2 ms to process 30 nodes in real-time, so it can also be run online according to the size of $Nodelist$ and transition of tasks. The experimental results are give in Section IV.

IV. Experiments and Performance Evaluation

In this section, an experiment using the FACT descriptor applied in the framework of topological mapping is introduced. We will report our test on functionality, time cost, and robustness separately. The real-time experiment is carried out on a differential drive robot with an omnidirectional camera on the top and with the assistance of built-in odometry. The experiment is carried out in a typical indoor office environment containing working office, printing room, coffee room, stairways , corridor, etc. The software is coded in C with the support of the OpenCV library. The white balance of the camera is automatically adjusted by referring to color from a certain area where a piece of white paper is sticked in the field of view.

A. The test on functionality

Our method will add new nodes into a topological map and recognize existing nodes automatically based on the FACT descriptors, while the mobile robot moves in a typical office environment. We set the frame rate to 1 fps, and acquire 1591 frames in total to take this test. It will also help us to try different algorithms based on the same sequence and clearly find the advantages and disadvantages. The test video covers the whole corridor shown in Fig.7 and six other different scenes such as printing room (trajectory in red), coffee room (brown and cyan), and stairways (in pink). The experimental result of the detected new nodes is shown along with the odometry information as reference. If the current scene is different from the last frame but exists in the $Nodelist$, it will not be shown in the figure. As we mentioned previously, we will remove all the nodes which look alike by performing an off-line refining method. The nodes which match each other for more than 70% were excluded. According to the experimental result, we can get a sketch map on the floor plan which is shown in Fig.7. The evaluation of the mapping result is in Table I. This

TABLE I
Evaluation of the mapping result

True Positive	False Positive	False Negative	# Valid nodes
9 (81.8%)	2 (18.2%)	2($\frac{2}{11+2}$=15.3%)	11

table shows that the FACT works well in the detection of new scenes, especially in detecting the the change of the type of scene. For example, the change from corridor to rooms are well identified, because these transitions usually contain significant color changes. But some of the scenes were not detected as new nodes, because of the similarity in feature, i.e. some places could be considered as the same appearance seen before. For instance, the upper one of the two missing scenes in an office which has no strong texture. This was filtered out during the refining stage. One feasible optimization would be to introduce corners or other simple features as supplements. It means besides the three features i.e. U, V and Width of regions, other features may be added to enhance the distinctive of *tags* in FACT. Moreover, the refinement of segmentation method will also help to get a more specified feature. These possible optimization will be carried out in our future work.

B. Test on time cost

The FACT descriptor should have low complexity in computation and above all fit the realtime requirement as we defined in the introduction. In order to certify the efficiency, the processing time for each frame must be measured. The chart of time cost by frame is shown in Fig. 8, which includes the time of unwrapping, segmentation, FACT extraction, node recognition and new nodes detection. 15 frames out of 1565f were rejected because the feature changed too fast

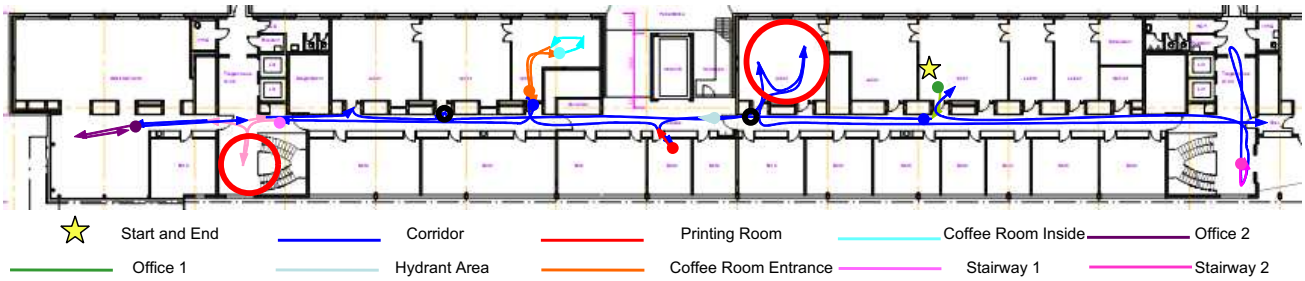


Fig. 7. A refined sketch. By performing an off-line refining, 9 nodes out of total 20 were excluded due to the similarity and isolated noise; 9 nodes out of the rest 11 were true positive result (the false positive are marked in black hollow circles); another 2 scenes were not identified (marked in red circles). All the labels are marked manually.

against the Hypothesis 1. The average processing time is 30.3 ms using an Intel CoreDual 3.0GHz CPU. The proportion of each phase is shown in Fig. 9.

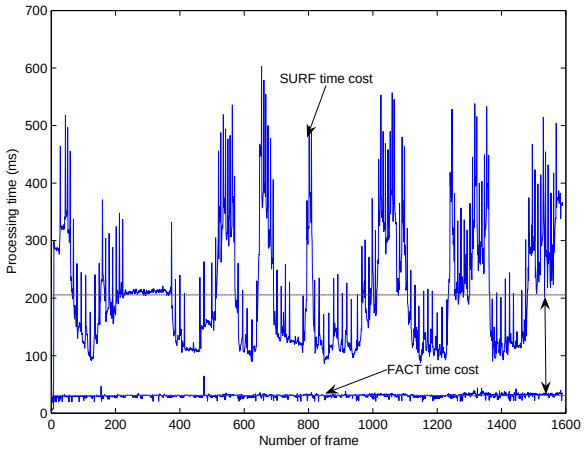


Fig. 8. The processing time using FACT and SURF. FACT: The time cost is slightly rising during the test, because the method is incremental and more new nodes will be detected and added to the $Nodelist$. The corridor in this experiment scenario is about 100-meter-long and finally 20 nodes are added. The unwrapping and segmentation will possess about 28.5ms in average, therefore the extraction and matching phases are as efficient as taking less than 7% processing time. The black line indicates the average of processing time. SURF: The processing time using SURF descriptor. We use the SURF method to ONLY detect the scene change without recognition function, which has already saved much time in matching stage. The green line indicates the average of the processing time of this method, and the black one is the average processing time using FACT descriptors. The arrow marks the difference of time cost between our method and keypoint-based method.

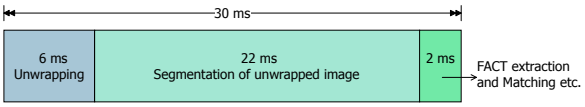


Fig. 9. A typical proportion of time cost in phases

As a comparison, a SURF descriptor based method was also tested. We take SURF as a representative of keypoint-based method, as it is faster in keypoint detection and matching than other candidates such as SIFT. By only detecting the scene change, without recognition function, the experiment was performed on the same video sequence. We abstracted SURF based keypoints and descriptors from each panoramic image. The ratio of $\frac{\#matched\ points}{total\#of\ keypoints\ in\ database}$ was employed to represent the matching score of every new frame. The overall processing time record is shown in Fig. 8. Because this experiment only detects the scene change, it did not need compare each node from the database. As the high dimension of the descriptor, the time cost for the matching phase varies from 9.1 to 268.6 ms (median value 52.2ms) depending on the number of keypoints of each node, on the same computer used in the previous test. We could imagine that if we join all the nodes together, the matching stage will take proportionally more time.

C. Test of robustness in dynamic environment

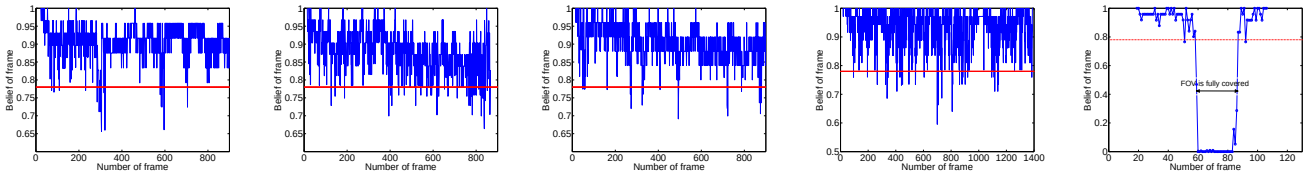
The method described in this work is supposed to omit the disturbance when less than 30% of the panoramic image in horizontal direction is affected. This characteristic is especially valuable when people may be walking around during the rove of the robot, after the topological map training. Ideally speaking, if more than 30% is covered by dynamic objects or people, the robot will take the current scene as an unknown area. The experiment for the robustness test is designed as below:

1) Move the robot to a typical indoor environment, and generate only one node for the test environment;
2) Invite number of people randomly walk within the field of view and take the log separately;
3) Change to another scene and repeat from step 1.

The experiment was taken in real-time and the test results are given in Fig. 10. The sub-figure (a)-(d) show the test results in the office room, in the corridor, in the stairway, and in the coffee room respectively. Because of the different color textures of these indoor environment, the robustness of FACT acted different. For the office environment, we had three people walking around consequently and they crowd together time after time.

The Fig.10(e) shows a result of an experiment on sensitivity, in which we suddenly cover the whole FOV and the algorithm shows its sensitivity to the sudden change, while the frame rate is around 20fps.

According to the test result, the FACT descriptor and the matching method have shown their ability in topological map and the robustness to dynamic objects in the environment.



(a) Office (b) Corridor (c) Coffee Room (d) Stairway (e) Test on fully covering the field of view

Fig. 10. The test result of the robustness of FACT. The red line indicates the threshold TH_{global} which defines the matching status of current image.

Above all, the time cost of FACT method is almost 7 times shorter than a typical keypoint based method, which will guarantee the mobile robot having more time slots in a real-time task.

V. Conclusions and Future Work

A. Conclusions

In this paper, we presented an adaptive and lightweight descriptor for omnidirectional vision named FACT (Fast Adaptive Color Tags). With the FACT descriptor, both scene-change detection and scene recognition can be fundamentally achieved. Above all, a node list for topological mapping can be produced in real-time.

The performance of this method was evaluated through a real-time experiment on a mobile robot with an omnidirectional camera. There are two important differences with previous work. The first one is that the nodes are generated only by the image appearance, without using other sensors such as ranger finders. The second one is that the extracted feature is based on color and geometric information, instead of other keypoints (like Harris, SIFT, or SURF). According to our test, the processing time is as fast as 30ms in average and the FACT descriptor is robust to local changes in the environment.

It should be noted that the functionality of FACT descriptor has been examined only in one certain indoor environment. The generalization may be solved by adjusting the two thresholds in the matching phase, supplementing features to the *tags* and refining the segmentation. Not withstanding its limitation, this study suggests that the descriptor based on segmentation and color could be used in scene recognition and topological mapping by costing comparatively short time and with robustness to occlusions and slight illumination changes.

B. Future Works

To optimize the FACT descriptor, one possible further work is to fuse other features such as corners or segments to enhance the uniqueness of the descriptor. Based on the FACT descriptor, it is feasible to build a navigation tree which can guide the robot transiting from one node to another adjacent node. The navigation could be achieved by comparing the difference of two image and minimize the error using a visual servoing approach [13].

References

[1] O. Booij, B. Terwijn, Z. Zivkovic, and B. Krose. Navigation using an appearance based topological map. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3927–3932, 2007.

[2] D. Floreano and F. Mondada. Evolution of homing navigation in a real mobile robot. *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, 26(3):396–407, 1996.

[3] T. Goedeme, T. Tuytelaars, L. Van Gool, D. Vanhooydonck, E. Demeester, and M. Nuttin. Is structure needed for omnidirectional visual homing? In *Computational Intelligence in Robotics and Automation, 2005. CIRA 2005. Proceedings. 2005 IEEE International Symposium on*, pages 303–308, 2005.

[4] P. Lamon, A. Tapus, E. Glauser, N. Tomatis, and R. Siegwart. Environmental modeling with fingerprint sequences for topological global localization. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 4, 2003.

[5] D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

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

[7] E. Menegatti, M. Zoccarato, E. Pagello, and H. Ishiguro. Image-based Monte Carlo localisation with omnidirectional images. *Robotics and Autonomous Systems*, 48(1):17–30, 2004.

[8] A.C Murillo, C. Sagüés, JJ Guerrero, T. Goedemé, T. Tuytelaars, and L. Van Gool. From omnidirectional images to hierarchical localization. *Robotics and Autonomous Systems*, 55(5):372–382, 2007.

[9] N. Otsu et al. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[10] C. Sagüés and JJ Guerrero. Visual correction for mobile robot homing. *Robotics and Autonomous Systems*, 50(1):41–49, 2005.

[11] D Scaramuzza, A Martinelli, and R Siegwart. A robust descriptor for tracking vertical lines in omnidirectional images and its use in mobile robotics. *International Journal of Robotics Research*, 2009. Special Issue on Field and Service Robotics.

[12] A. Tapus and R. Siegwart. Incremental robot mapping with fingerprints of places. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 2429–2434, 2005.

[13] N. Winters, J. Gaspar, G. Lacey, and J. Santos-Victor. Omni-directional vision for robot navigation. In *IEEE Workshop on Omnidirectional Vision*, pages 21–28, 2000.

[14] L Zhao, R Li, T Zang, L Sun, and X Fan. A Method of Landmark Visual Tracking for Mobile Robot. In Xiong, C and Liu, H and Huang, Y and Xiong, Y, editor, *Intelligent Robotics and Applications, PT I, Proceedings*, volume 5314 of *Lecture Notes in Artificial Intelligence*, pages 901–910, 2008.