Conference Paper

Scene recognition with omnidirectional vision for topological map using lightweight adaptive descriptors

Author(s):
Liu, Ming; Scaramuzza, Davide; Pradalier, Cédric; Chen, Qijun; Siegwart, Roland

Publication Date:
2009

Permanent Link:
https://doi.org/10.3929/ethz-a-010035843

Originally published in:
http://doi.org/10.1109/IROS.2009.5354131

Rights / License:
In Copyright - Non-Commercial Use Permitted
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.

In this work, we employ the unwrapped image to simplify the extraction of descriptors. 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 segmented image, the matching algorithm, and the generation of the topological map. The evaluation and experiments can

This work was supported by CSC (China Scholarship Council) and Robots@home STREP Project IST-6-045350
Ming Liu is with CEIE of Tongji University and Autonomous Systems Laboratory in ETH Zurich. ming.liu@mavt.ethz.ch

be found in Section IV, where we also make a comparison between the computational cost of our approach and that of the keypoint-based approach.

II. SEGMENTATION OF THE PANORAMA

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

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

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.

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 \(D\) is shown in Eq. (1). Each column in the descriptor is named Tag.

\(^1\)According to our experiment, \(N\) is usually smaller than 100 and greater than 20.
\[
D = \begin{pmatrix}
U_1 & U_2 & \ldots & U_N \\
V_1 & V_2 & \ldots & V_N \\
W_1 & W_2 & \ldots & W_N
\end{pmatrix}
\] (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.

![Base image (D₁)](image1)
![Current image (D₂)](image2)

\[\text{Dis}_1 = \text{Min}(\text{Dis}_{1,1}, \text{Dis}_{1,2}, \ldots, \text{Dis}_{1,j})\]

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 \( \text{Dis}_1 \) is selected as represent of the region. If \( \text{Dis}_{1,i} \) 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.

![Fig. 5. A schematic diagram of Test 1 during the matching process.](image3)

In order to find the candidate match for a given descriptor, we devised the three tests listed below.

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.

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

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

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 results of two consecutive frames are lower than the predefined threshold $T_{H\text{globe}}$ (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 $I_m(n)$ 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 refining process is similar to the matching process except that time we use a lower threshold $T_{H\text{refine}}$ (in this work $T_{H\text{refine}} = 70\%$) to enable tolerance until a certain extent. We start from the last node in the $Nodelist$ and compare it with all the previous nodes using the descriptor matching method explained in subsection B. If there are node(s) which can be matched with the current node, fitting the $T_{H\text{refine}}$, the current node will be removed from the $Nodelist$. The process runs repeatedly until it reaches the base node $Node_1$. The list of all the remaining valid nodes, along with the refined $FACTlist$, is the output of our method. 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.

#### A. 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 shows that the FACT works well in the detection of new scenes, especially in detecting 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. This kind of problem is related to the sensitivity of the matching method, because there are two thresholds which define the matching result of one region and the existing nodes during the online exploring process, while a third threshold is used to refine the $Nodelist$. Changing these thresholds significantly affects the result. In the experiment, we chose the empirical value $3e^{-4}$ as the distance threshold in a $10\times 10$ U-V space and 78% as the threshold of the matched ratio. Both the two thresholds are compromised results according to our test environment. On one hand, the threshold should be high enough to eliminate false positive; on the other hand, the higher threshold will cause more meaningless nodes. By finding a good compromise for these three values, the sensitivity could be modified accordingly. Another possible optimization would be to introduce corners.

![Fig. 6. The pattern of Nodelist and FACTlist](image-url)
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 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.

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 matched points was employed to represent the matching score of every new frame. The overall processing time record is shown in Fig. 10. 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. Take Fig. 11 for instance, the people outlined by green is not supposed to influence the recognition result comparing with the left image taken at the corridor. The important fact should notice that the matching result of image is based the number of regions, not the matching area. This changes when dynamic objects/people occludes the vertical edge. Therefore, the experiment 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 matched points was employed to represent the matching score of every new frame. The overall processing time record is shown in Fig. 10. 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.

CONFIDENTIAL. Limited circulation. For review only.
regions in color. Another reason is the space is bigger than that in the coffee room or office room, so during the test, people could walk farther compared with other places. Therefore, sometimes only a small part of the FOV (Field Of View) is occluded, which also makes the result look better. The Fig.12(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. 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. When a reliable topological map is autonomously produced, the next task is visual navigation. Based on the FACT descriptor, it is possible 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].

Another possible direction is the fusion of other sensors besides omnidirectional vision, e.g., stereo vision, which will provide more precise image than hyperbolic mirrored camera and the distance information will help to correct the segmentation and awareness of obstacles, while staying comparatively low cost. An interesting application could be locating the robot in a node from a topological map through FACT, then creating an object-based map on a lower but more precise level.

**REFERENCES**


