International Journal of Control and Automation
Vol.8, No.5 (2015), pp.281-292
http://dx.doi.org/10.14257/ijca.2015.8.5.26

Using Visual Feature and Geometric Constraints for Robot Localization

Sangyun Lee[1], InPyo Lee[2], Changkyung Eem[3], and Hyunki Hong[4]

[1, 2] Department of Imaging Science and Arts, GSAIM, Chung-Ang University,Seoul,Korea
88leesy@gmail.com[1], vjdlsvy@nate.com[2]
[3] College of ICT Engineering, Chung-Ang University, Seoul, Korea
richardeem@gmail.com
[4] School of Integrative Engineering, Chung-Ang University, Seoul, Korea
honghk@cau.ac.kr

Abstract

This paper presents a novel method to generate an index word for the topological map in robot localization. Previous studies extract only appearance features from an input image to match the visual words of the model images. However, the localization performance is much affected by the miss or false matches. First, we segment a robot navigation environment into the structural planes using 3D depth data. We obtain both the surface normal vectors of the structural planes and visual features in the model image, which are compared with those of an input request image in the voting approach. The experimental results show the voting performance is improved by taking into account the spatial distribution of the features.

Keywords: *robot localization, topological map, 3D-depth data, voting approach*

1. Introduction

In autonomous robot navigation, Simultaneous Localization and Mapping (SLAM) is a technology to localize robots and build a map in an unknown environment. In addition to the use of traditional lasers and radars, cameras have been competitive alternatives in SLAM owing to their low cost and rich information content [1]. Recent research extends to building a 3D geometric map of an office environment using a ground mobile robot equipped with a Microsoft kinect camera [2-3].

The SLAM method generates a sparse cloud of point features that is suitable for estimating the pose of the camera but makes little effort to extract any geometric understanding from the map [1, 4]. This paper deals with the problem of building the planar structure of a robot navigation environment and estimating the localization on a qualitative basis. We assume that the surrounding environment where the robot drives is constructed generally with many structural planes, such as walls and ceilings. The image patch is a small region with characteristic appearances and the scene is represented with a collection of patches on the structural plane. Because structural information that included straight lines and planes is abundant in man-made environments, they are used commonly in various vision tasks [4-6].

Localization to estimate a precise robot position is a basic requirement for robotic application. This capacity in complex environments relies on a map which can be either given to the robot, or learned while the robot discovers its surroundings. Navigation systems use either topological or metric maps. A topological method enables to tell where the robot is present, and is used to initialize a metrical localization [7]. Previous localization and map-learning systems employ a visual word, a small patch on the image,

ISSN: 2005-4297 IJCA

carrying any kind of interesting information in any feature space such as color patches or KLT features [8-11].

In order to improve the recognition performance in the topological map, the proposed method takes into account both the visual appearances and their geometric constraints. First, we extract the structural planes from dense or sparse 3D points using Random Sample Consensus (RANSAC). To tell where the robot is located, we find image features that are located on the planes, enabling a survey of every structured visual word in which it appears. In addition, the proposed method includes the semi-local constraint of visual features [12], representing their geometric distribution among the structural planes in the scene. In the final step, we examine the voted features on the plane considering their relative positions in the Left-to-Right and Up-to-Down (LRUD) direction to improve the voting performance.

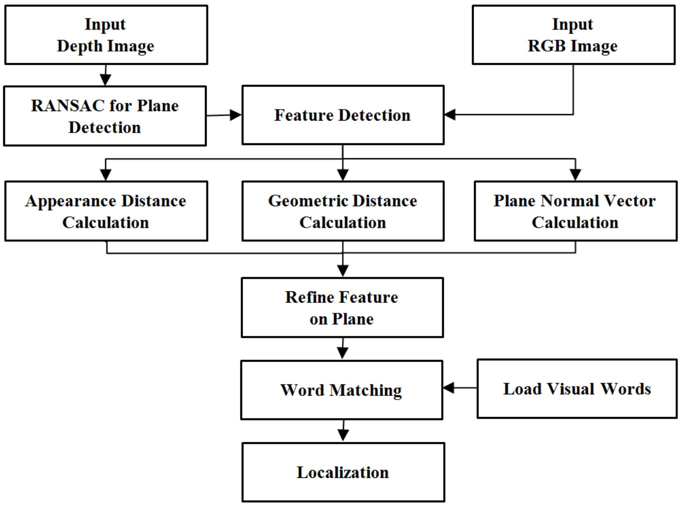


Figure 1. Proposed Flow Chart

2. Related Works

The automatic discovery of higher level structures such as planes in unprepared environments is an important step towards enabling more complex interactions between real and virtual objects for application of AR. Chekhlov employs RANSAC to search for planes in the point cloud of the SLAM map and the best-fit plane is determined with the inlying points from the plane hypothesis with the most consensuses [4]. The plane structural components are augmented into the SLAM state, maintaining inherent uncertainties via a full covariance representation [13]. However, if the camera observes planes that are far away from the calibration target, the obtained planes would have greater uncertainty.

Previous methods to automatically detect planar regions are based on the comparison of transfer errors of homography, which makes them very sensitive to the choice of a discrimination threshold [14-16]. Because the homography error does not reflect precisely in the degree of the co-planarity, it is difficult to determine the reasonable threshold value in various situations. In addition, an iterative voting scheme like Least Median of Squares (LMS) to identify coplanar subsets of the feature set and refine the homography estimates is too slow for real-time operation [14]. Simon proposed a method to detect and reconstruct planar surfaces using hough transform and a reference plane for AR applications [17]. The method has the following limitations: 2D polygon corresponding to the reference plane was outlined by user input in the first frame and the reference plane has to be visible through the whole sequences.

Nasir *et al.* describe the robot system using a kinect sensor to create a geometric feature like 3D plane based map in an indoor environment [2, 3]. To extract multiple 3D planes from the point cloud data, two methods use Hough transformation and RANSAC respectively. However, if the kinect would look straight forward into the moving direction of the robot, most information would be lost since kinect has limited depth range and the parts of the corridor have generally longer lengths. The kinect sensor acquires enormous amounts of data, which are a challenging problem in real-time applications. Furthermore, because the kinect sensor uses an infrared band, it is mainly restricted to indoor application.

Jurie et al represents images as a set of unordered elementary features (the words) taken from a dictionary or codebook [18]. Using a given dictionary, the classifier is based on the frequencies of the words in an image. The words are local image features, which can be represented with image patches, histograms of gradient orientations or color histograms. Dictionary building and classifier training are performed on database images through off-line learning.

Filliat presents a visual localization and map-learning system using topological maps [7]. From topological maps the robot can recognize the room it is in, but cannot obtain its metrical position in precise. When building the dictionary and gathering data for the classifier, various image features including Scale Invariant Feature Transform (SIFT) description, local color histogram and local normalized gray level histogram are used. Because the method depends on mainly the image features, however, its performance would be much affected by the miss and/or false matches.

The topological map of the surrounding environment is used widely for visual odometry system using an omni-directional camera [10-11, 19]. Although the omni-directional camera has a wide total view angle, it is generally mounted in the upward direction. Therefore, the surrounding environments such as walls are projected into the small image areas on field of view, causing it to become difficult to detect and establish correspondence with feature points. In addition, only the appearance information is used for robot localization, and there is no consideration of the scene structure. Tapus *et al.* detect the vertical edges and color patches of the panoramic image, and extract the corner features using multiple laser finders and an omni-directional camera [10]. The features are re-ordered over the sequence according to their angular positions. The system which is equipped with multiple sensors including laser sensors and an omni-directional camera has to combine the range data and the visual features. Since the surrounding environments are radially projected by the omni-directional camera, the camera motion causes the image features, such as color patches, to change.

3. Proposed Method

This paper presents a novel localization method using image features and their semi-local constraints among structural planes. The proposed method is applicable for both the stereo camera and the kinect sensor to capture 3D data for use in a robot navigation environment. At first, RANSAC is used to extract the structural plane and its surface normal information from dense or sparse 3D datum. We obtain feature points and describe their appearance information with RGB color histogram or Binary Robust Invariant Scalable Keypoint (BRISK) [20].

Both the appearance and the geometric features on the structural plane are indexed by the index word of the image model, which is compared with that of the request image. Using the voting algorithm enables us to select the image of the base, which is most similar to the request. Furthermore, the semi-local constraints are examined in matching process to improve the voting performances. In order to deal with mismatches as well as outliers, we compare the relative order relation (left, right, up and down) of the matching candidates with that of the key points. Figure 2 shows the model images by kinect and the stereo vision system mounted on the robot in an indoor and outdoor environment.

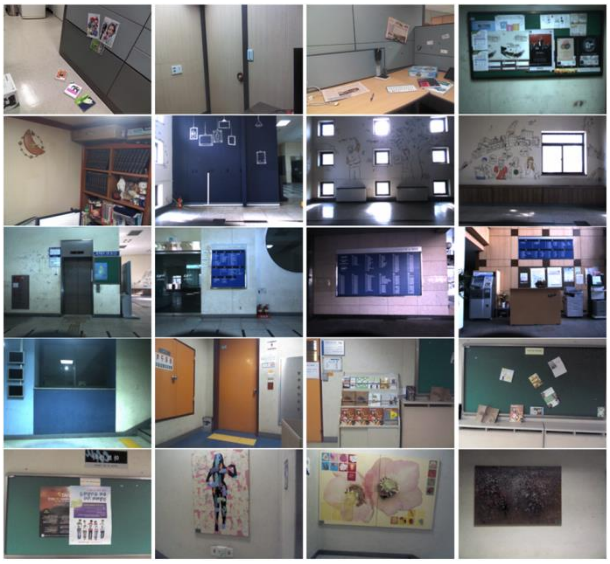


Figure 2. Model Images (Scenes 1~20, from left top)

3.1. Plane Segmetation and Visual Word Generation

By applying RANSAC approach to dense or sparse 3D points, we obtain the main planes where many 3D points are distributed. Principal Component Analysis (PCA) enables us to compute a surface orientation representing the eigenvector of the smallest eigenvalue, from 3D points of the main planes [10]. Backface culling is used to determine the visual surface normal of the structural plane according to the viewpoint of the camera. Then, we group the features on the planes into 3D key points, and their descriptors are stored for building the index words of the model and voting the request images.

The proposed algorithm measures both similar appearance and 3D distance between the features, which is used to cluster the scene objects with no additional assumptions [21]. In equation (1), dGeom is the normalized distance representing the geometric information of the feature. xi and xj are the neighboring feature positions in 3D space, and c is the camera position, respectively.

$$d_{Geom}(i,j)^2 = \frac{\left\|x_i - x_j\right\|^2}{\max(\left\|x_i - c\right\|^2, \left\|x_j - c\right\|^2)}, \tag{1}$$

$$s(i,j) = \exp(-\frac{d_{RGB}(i,j)^2}{\sigma_{RGB}^2} - \alpha \frac{d_{Geom}(i,j)^2}{\sigma_{Geom}^2}). \tag{2}$$

Each feature’s appearance is characterized using a 1D RGB histogram with 16 bins per channel computed over 15×15 pixel image patch around the feature position. Equation (2) represents the similarity s(i, j) between the feature i and j, and we compute the color histogram distance dRGB(i, j). σRGB and σGeom are the scaling parameters for setting to the same percentage of the range of the respective distance functions; α is a relative weight of the geometric term. Figure 3 shows the disparity map by the stereo system [22], detected features and the segmented two planes with red and green colors.

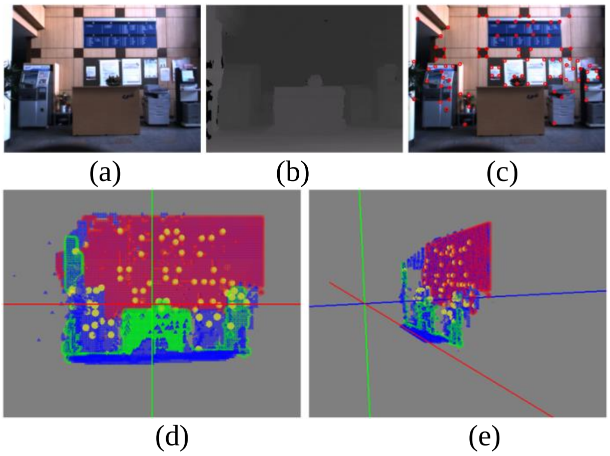


Figure 3. (a) Input Image (b) Disparity Map (c) Detected Points (d) and (e) Segmented Planes

The index word with a fairly ordered appearance is useful to vote for the place where the robot is. Because the previous approach depends on the image features mainly, however, its performance would be much affected by the miss and/or false matches [7]. On the contrary, the proposed method examines the visual features and the geometric information: semi-local constraints and the relative consistency of the key points in the left, right, up and down direction.

3.2. Matching and Localization

The idea of voting algorithm is to sum the number of times each word is selected [7]. The scene that is chosen most often is considered to be the best match. The proposed method builds a base with the model images, which are represented as the features of the key points and their geometric constraints. In other words, we obtain the key points and their geometric information of the model image in the learning stage, which compare with those of the request image for robot localization.

Figure 4 shows how to encode the key points and their spatial relationship between two structural planes. The surface normal vectors of two planes with green and red colored key points and their in-between angles θ1 and θ2 are represented, respectively. An in-between angle between the structural planes is computed using the dot product of two surface normal vectors.

Equation (3) calculates the feature similarity χij2 in the color histogram between i and j. k and $\vec{H}(k)$ are the kth bin histogram and the averaged histogram of i and j, respectively. The small χij2 distance value means that the input feature i is similar to the visual word j of the model. In order to deal with the false matches, we set the threshold value to a value of 0.4 experimentally. When a BRISK descriptor is used instead of the color histogram, we are able to determine efficiently whether the features of the request would be matched with the key points of the request, using the binary hamming distance.

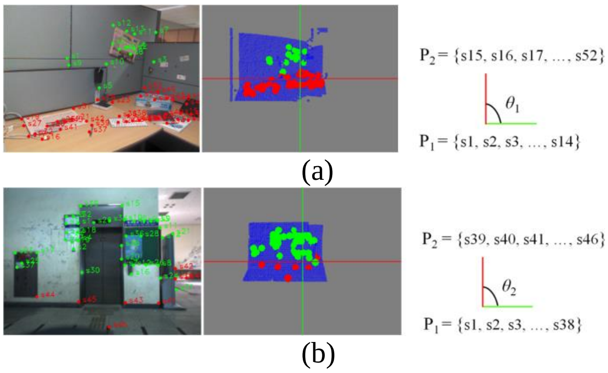
$$\chi_{ij}^2 = \chi^2(\vec{H}_i, \vec{H}_j) = \sum_k \frac{(\vec{H}_i(k) - \vec{H}(k))^2}{\vec{H}(k)}, \tag{3}$$

$$k_{word} = \arg\min_k (\alpha(1 - \frac{N_{c\_pass}}{N}) + \beta\left(\frac{1}{n}\sum_{i=1}^{n} \frac{|m_i - r_i|}{180}\right)), \qquad (n \ge 1) \tag{4}$$

The word k_{word} minimizing the measure of both the appearance and geometric distances is computed using equation (4). In the first term, N is the total number of the voting process and Nc_pass is the number of the features passed in the visual similarity test as an equation (3), respectively. The second term examines the geometric similarity between the request and the model image using the absolute difference of the in-between angles among the structural planes. Here mi and ri are the in-between angle of the model and that of the request. n represent the number of the in-between angles in the request.

The different angles between the neighboring planes is ranged within 0-180 degrees, because the normal of the visible surface is considered. For example, when two planes are facing each other, the angular difference is 180 degrees. The structural geometric term is divided with 180 for the normalization. We control the relative contribution of two terms with α and β.

From the experimental results, we found that better performances are obtained in case both the appearance feature and the geometric constraints are considered equally, so the weights of the two terms are set to 0.5 and 0.5.

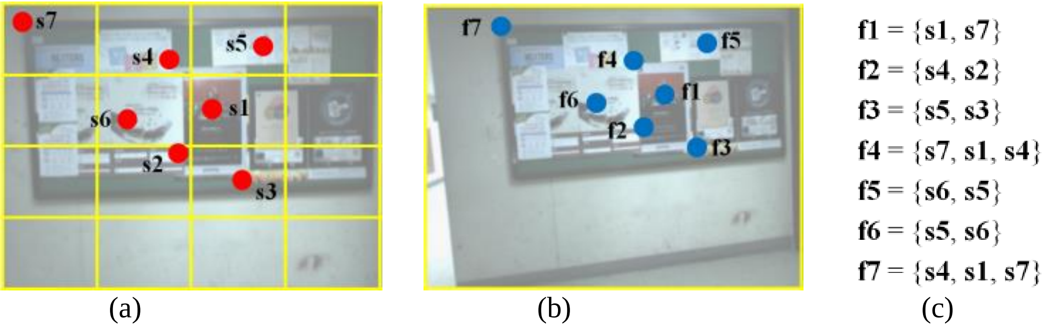

$P_2 = \{s15, s16, s17, \ldots, s52\}$
θ_1
$P_1 = \{s1, s2, s3, \ldots, s14\}$
$P_2 = \{s39, s40, s41, \ldots, s46\}$
θ_2
$P_1 = \{s1, s2, s3, \ldots, s38\}$

(a)

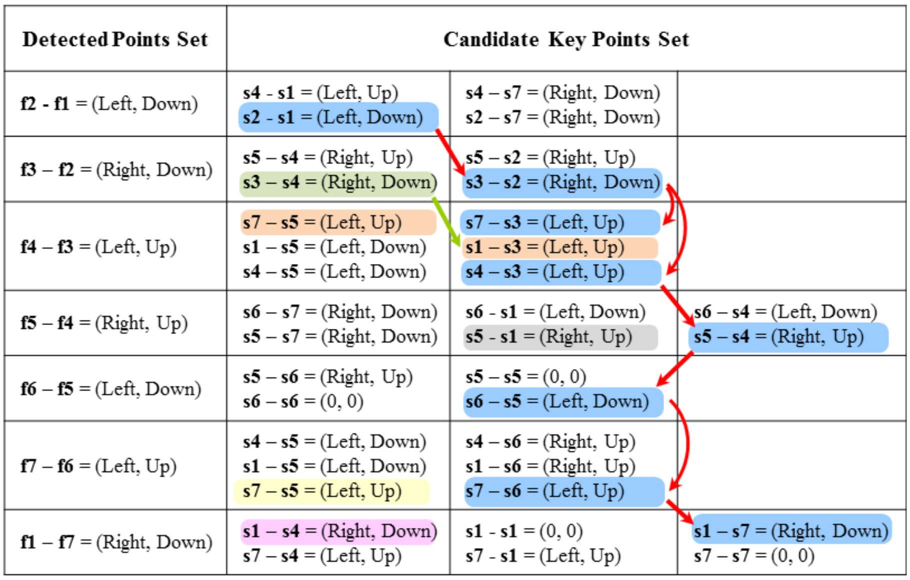
(b)

Figure 4. Key Points and their Geometric Relation between the Structural Planes in the Scene 3 and 9 ((a) and (b))

When the number of the plane in the image is 1, meaning n = 0 in equation (4), the proposed method examines the relative order of the key points on the structural plane instead of the in-between angle. In addition, using this verification process enables us to decrease the effects of the false matching and the missing features. More specifically, when matching the input features and the key points of the model, falsely matched pairs occur necessarily in most cases. In order to solve the false matching problem, we examine the relative order in Left-to-Right and Up-to-Down between visual features as Figure 5. We assume that there is no rotation about the front-to-back axis called roll because 3D sensor is mounted on the mobile robot navigating on the ground.



(a) (b) (c)



Detected Points Set	Candidate Key Points Set		
f2 - **f1** = (Left, Down)	**s4** - **s1** = (Left, Up) **s2** - **s1** = (Left, Down)	**s4** – **s7** = (Right, Down) **s2** – **s7** = (Right, Down)	
f3 – **f2** = (Right, Down)	**s5** – **s4** = (Right, Up) **s3** – **s4** = (Right, Down)	**s5** – **s2** = (Right, Up) **s3** – **s2** = (Right, Down)	
f4 – **f3** = (Left, Up)	**s7** – **s5** = (Left, Up) **s1** – **s5** = (Left, Down) **s4** – **s5** = (Left, Down)	**s7** – **s3** = (Left, Up) **s1** – **s3** = (Left, Up) **s4** – **s3** = (Left, Up)	
f5 – **f4** = (Right, Up)	**s6** – **s7** = (Right, Down) **s5** – **s7** = (Right, Down)	**s6** - **s1** = (Left, Down) **s5** - **s1** = (Right, Up)	**s6** – **s4** = (Left, Down) **s5** – **s4** = (Right, Up)
f6 – **f5** = (Left, Down)	**s5** – **s6** = (Right, Up) **s6** – **s6** = (0, 0)	**s5** – **s5** = (0, 0) **s6** – **s5** = (Left, Down)	
f7 – **f6** = (Left, Up)	**s4** – **s5** = (Left, Down) **s1** – **s5** = (Left, Down) **s7** – **s5** = (Left, Up)	**s4** – **s6** = (Right, Up) **s1** – **s6** = (Right, Up) **s7** – **s6** = (Left, Up)	
f1 – **f7** = (Right, Down)	**s1** – **s4** = (Right, Down) **s7** – **s4** = (Left, Up)	**s1** - **s1** = (0, 0) **s7** - **s1** = (Left, Up)	**s1** – **s7** = (Right, Down) **s7** – **s7** = (0, 0)

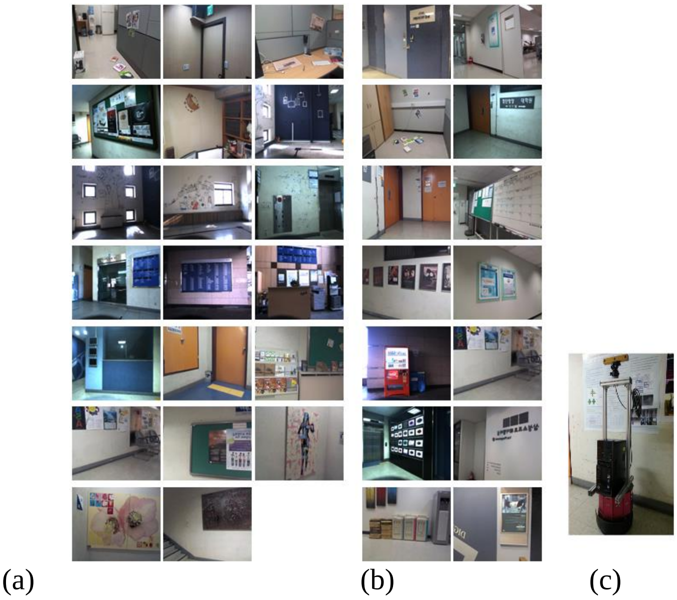
(d)

Figure 5. (a) Key Points of the Model Image (b) Detected Points of the Request (c) their Matching Candidates on the Structural Plane (d) LRUD Vertification Process

As shown in Figure 5, the method checks out whether the image coordinates difference between the neighboring feature points in the requested image satisfies the directional order–left-to-right and up-to-down–of the matched key points in the model. The key points set satisfying the image, coordinate difference is saved as the connection list. For example, the detected point f2 is located on the left and down side of f1. By examining the matched candidate key points set satisfying the relative position order, s2 and s1 are found as Figure 5 (d). In the next step, there are two candidates with the same relation between f3 and f2: s3 - s4 and s3 - s2. The first set is a new connection and the second is linked with the previous connection. Several candidate sets may be generated in every step, but they are hard to satisfy steadily the geometric relation among the subsequent sets.

Figure 5 (d) shows the connection in the sky blued boxes has the longest length, representing the matched structural plane of the model. If there would be no correct correspondence among the matched candidates, the list is disconnected. In this case, we verify another combination pair of the points such as f3 - f1 and f5 - f1. The consideration is helpful for dealing with the situation: the connection list with the longest length is more than two and there are many false matching points.

In the final step, we compute the sum of the distances from the main key points near a center position to another point in the model. After we compared the distance sum of the request with that of the model, the connection with the minimum distance is determined as a true vote result. The proposed method to evaluate the geometric consistency of the features between the request and the model is applicable to index and match visual words in image retrieval in spite of many falsely matching or missing features.



(a) (b) (c)

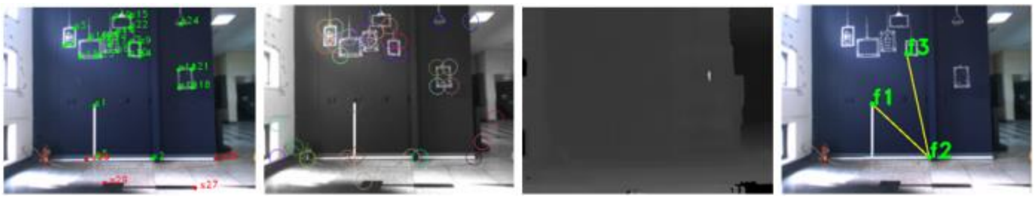
Figure 6. (a) Input Request Images (True Positive) (b) True Negative Image (c) Stereo System Mounted on Mobile Robot

4. Experiments and Discussion

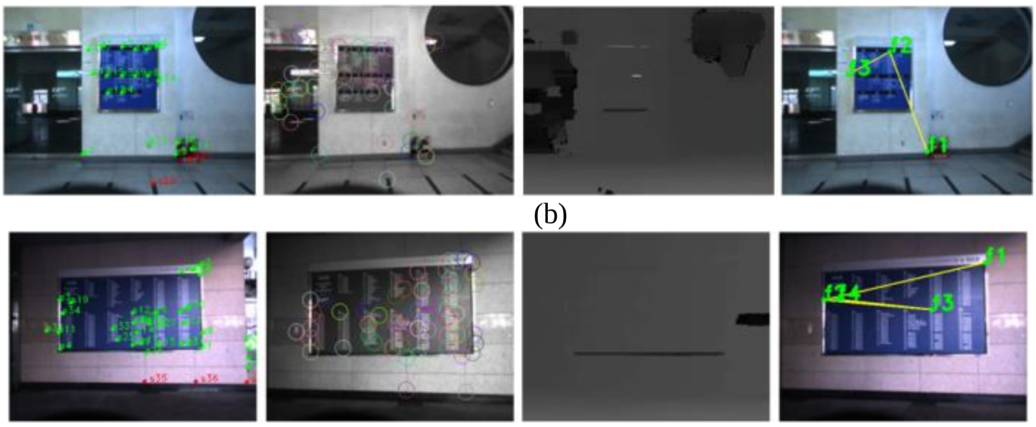
The computational equipment used includes an Intel Core (TM) i7-3770 3.4GHz and 8GB RAM with NVIDIA GTX680 graphics card. The stereo images are captured by a Bumblebee 3 from Point Grey Inc. at a rate of 15 fps (frame per second), and kinect senses 3D depth in the scene at 30 fps. The stereo matching is implemented on GPGPU architecture at a rate of 12~15 fps, processed in parallel threads [22]. There are 20 scenes in the robot navigation environment that are learned and matched to tell where the robot is located. In order to compare the voting performances, 379 input images of the scene (true positive) and 209 negative images with the same appearance features as the target scene are used as Figure 6(a) and (b).

Figure 7 shows the key points on the structural planes, the detected features in the input, the disparity map and the verified key points, respectively. In most scenes, 60-70 key points are indexed in the learning stage and 3-5 points among them are left in the voting approach finally as shown in figure 6.

In the input scenes (Figure 2), we compare the recognition results from the previous method [7] and the proposed system as Figure 8. The proposed method describes the key points in the model image using the color histogram or BRISK. Using BRISK descriptor enables us to obtain better performances in most of input images than using the color histogram. However, the method using BRISK is affected to a large extent by the image characteristics, such as a regular and repeated appearance pattern as scenes 5 and 11. Furthermore, few key points are extracted on the structural planes in scene 5 with little textured surfaces. On the contrary, the method using the color histogram is difficult to discriminate scene 19 because its color distribution is similar to that of another model.



(a)



(b)

(c)

Figure 7. (a-c) In Scene 6,10 and 11, Key Points on the Structural Plane Detected Features, Disparity Map, and Verified Key Point (from left)

Table 1. Computation Time in Leaming and Matching Step

Leanring Step			Voting Step		
Methods		Times (msec)	Methods		Time (msce)
Coner detection	GoodFeature To Track[22]	6.818	Feature matching	RGB Histogram	6.379
	AGAST(in BRISK[19])	1.629		BRISK	4.558
3D Point Cloud generation		0.055	Visual word matching		11.634
Plane segmentation (RANSAC)		4.062			
PCA		0.129	RLDU verification		10.313
Feature clustering		8.884			
Descriptor eneration	RGB Histogram	559.213			
	BRISK	1.536			

Average recognition rates of the previous and the proposed method (color histogram and BRISK) are 41.52%, 61.73%, and 74.94%, respectively. The voting performances are improved by considering the spatial distribution of the features.

Table 1 provides computation times of the proposed method using color histogram or BRISK. In learning stage to build visual and geometric words, two methods take 579.16 msec and 7.41 msec. More specifically, the approach with BRISK have the same processes excepting corner detection, features clustering and description generation. The computation results show that it takes most of computation times to generate the color histogram of the key point. In the voting stage to match the request with the models, they take 16.69 msec and 14.87 msec, respectively. In the case of BRISK descriptor, we employ the binary hamming distance with computational efficiency. Previous method using appearances only takes 11.63 msec. However, evaluating the geometric consistency between the request and the key point enables us to perform a precise matching in spite of many falsely matching or missing features.

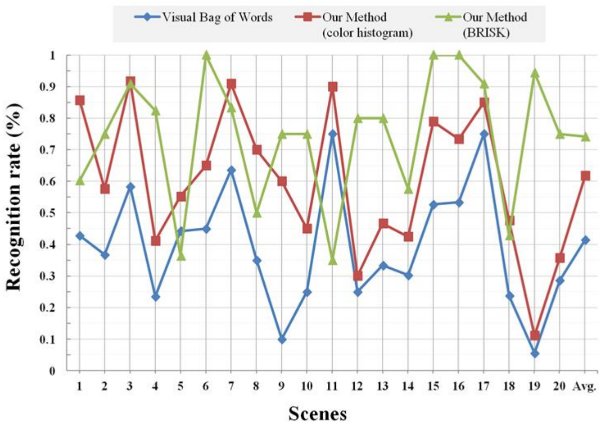


Figure 8. Comparison of Voting Performances

5. Conclusion

This paper presents a novel method to generate the index words for the topological map in robot localization problem. Because in using only the appearance features, the previous research is much affected by the miss and false matches. The proposed algorithm segments a navigation environment into the structural planes using 3D depth data. Then, we are able to obtain both visual features and their surface normal vectors in the model, which are compared with those of an input image in the voting approach. In order to improve voting performance, we make the verification using the geometric constraints: the angle between the structural planes and LRUD consistency examination of the key points.

Acknowledgements

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 (2013R1A1A2008953).

References

[1] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “MonoSLAM: real-time single camera SLAM”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 6, **(2007),** pp. 1052-1067.

[2] A. K. Nasir, C. Hille, and H. Roth, “Plane Extraction and Map Building Using a Kinect Equipped Mobile Robot”, In Workshop on Robot Motion Planning: Online, Reactive and in Real-time, IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, **(2012),** pp. 7-12.

[3] S. Lee, C. Eem and H. Hong, “Robot localization method based on visual features and their geometric relationship”, Proceedings of the 10th International Workshop on Infromation Technology and Computer Science, pp. 46-50, **(2015)**, April 15-18; Jeju Island, Korea

[4] D. Chekhlov, A. P. Gee, A. Calway, and W. M. Cuevas, “Ninja on a plane: automatic discovery of physical planes for augmented reality using visual SLAM”, In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, **(2007),** pp. 153-156**.**

[5] M. I. A. Lourakis, A. A. Argyros, and S. C. Orphanoudakis, “Detecting planes in an uncalibrated image pair”, Proceedings of BMVC, **(2002),** pp. 587-596**.**

[6] G. Simon, A. W. Fitzgibbon, and A. Zisserman, “Markerless tracking using planar structures in the scene”, Proceedings of ISMAR, **(2000),** pp. 120-128**.**

[7] D. Filliat, “A Visual Bag of Words Method for Interactive Qualitative Localization and Mapping”, Proceedings of IEEE Int’l. Conf. on Robotics and Automation, **(2007),** pp. 3921-3926**.**

[8] J. Yang, Y. Jiang, A. G. Hauptmann and C. Ngo, “Evaluating Bag-of-Visual-Words Representations in Scene Classification”, Proceedings of Int’l. Workshop on Multimedia Information Retrieval, **(2007),** pp. 197-206**.**

[9] A. Angeli, D. Filliat, S. Doncieux, and J. Meyer, Fast and Incremental Method for Loop-Closure Detection Using Bags of Visual Words. IEEE Transactions on Robotics, vol. 24, no. 5, **(2008),** pp. 1027-1037**.**

[10] A. Tapus, and R. Siegwart, "Incremental Robot Mapping with Fingerprints of Places", Proceedings of IEEE/RSJ Int'l. Conf. on Intelligent Robots and System, **(2005**), pp. 2429-2434.

[11] P. E. Rybski, F. Zacharias, J. Lett, O. Masoud, M. Gini, and N. Papanikolopoulos, "Using visual features to build topological maps of indoor environments", Proceedings of IEEE Conf. on Robotics and Automation, **(2003),** pp. 850-855**.**

[12] C. Schmid and R. Mohr, "Local Gray Value Invariants for Image Retrieval", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 5, **(1997),** pp. 530-535**.**

[13] A. P. Gee, D. Chkhlov, W. Mayol, and A. Calway, "Discovering Higher Level Structure in Visual SLAM", IEEE Transactions on Robotics, vol. 24, no. 5, **(2008),** pp. 980-990**.**

[14] M. I. A. Lourakis, A. A. Argyros, and S. C. Orphanoudakis, "Detecting Planes in an Uncalibrated Image Pair", Proceedings of BMVC, **(2002),** pp. 587-596**.**

[15] Q. He, and C. H. Chu, "Planar Surface Detection in Image Pairs Using Homographic Constraints", Lecture Notes in Computer Science, vol. 4291, **(2006),** pp. 19-27**.**

[16] A. Amintabar, and B. Boufama, "Homograhpy-Based Plane Identification and Matching", Proceedings of ICIP, **(2008),** pp. 297-300**.**

[17] G. Simon, "Automatic Online Walls Detection for Immediate Use in AR Tasks", Proceedings of ISMAR, **(2006),** pp. 39-42**.**

[18] F. Jurie, and B. Triggs, "Creating Efficient Codebooks for Visual Recognition", Proceedings of Int'l. Conf. on Computer Vision, **(2005),** pp. 604-610**.**

[19] W. L. D. Lui, and R. Jarvis, "A Pure Vision-Based Approach to Topological SLAM", Proceedings of IEEE/RSJ Int'l. Conf. on Intelligent Robots and System, **(2010),** pp. 3784-3791**.**

[20] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints", Proceedings of ICCV, **(2011),** pp. 2548-2555**.**

[21] A. Angeli, and A. Davison, "Live Feature Clustering in Video Using Appearance and 3D Geometry", Proceedings of BMVC, **(2010),** pp. 41.1-41.11**.**

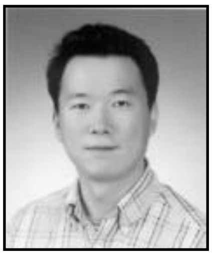
[22] B. Nam, S. Kang, H. Hong, and C. Eem, "Stereo System Based on a Graphics Processing Unit for Pedestrian Detection and Tracking", Optical Engineering, vol. 49, no. 12, **(2010),** pp.127203:1-9**.**

[23] J. Shi and C. Tomasi, "Good features to track", Proceedings of CVPR, **(1994),** pp. 593- 600**.**

Authors

SangYun Lee, he received his BS degree in Department of Electronic Engineering from Hoseo University, Asan City, Korea, in 2014. He is currently his MS degree in the Department of Imaging Science and Arts, Graduate School of Advanced Imaging Science, Multimedia and Film (GSAIM) at Chung-Ang University, Seoul, Korea. His research interests include augmented reality and computer vision.

InPyo Lee, he received his BS degree in computer game engineering from Korea Polytechnic University, Koera, in 2010. He received MS degree from the Department of Imaging Science and Arts, GSAIM, Chung-Ang University, in 2013. He is currently working for SamSung Electronics, Inc. Korea. His research interests include computer vision.



Changkyoung Eem, he received his BS, MS, and PhD degrees in Electronic Engineering from Hanyang University, Seoul, Korea in 1990, 1992, and 1999, respectively. From 1995 to 2000 he worked at DACOM R&D Center as a research engineer. In 2000, he founded a network software company, IFeelNet Co. and worked for Bzweb technologies as a CTO from 2006 to 2009 in the USA. From 2009 to 2011, he was an invited professor of the Department of Electrical Engineering in KAIST, Daejun, Korea. Since 2014, he has been an industry-university coorperation professor in College of ICT Engineering at Chung-Ang University, Seoul, Korea. His research interests include augmented reality and computer vision.

Hyunki Hong, he received his BS, MS, and PhD degrees in Electronic Engineering from Chung-Ang University, Seoul, Korea, in 1993, 1995, and 1998, respectively. From 1998 to 1999 he worked as a researcher in the Automatic Control Research Center, Seoul National University, Korea. From 2000 to 2014, he was a professor in the Department of Imaging Science and Arts, Graduate School of Advanced Imaging Science, Multimedia and Film at Chung-Ang University. Since 2014, he has been a professor in the School of Integrative Engineering, Chung-Ang University. His research interests include computer vision and augmented reality.