2012 IEEE International Conference on Robotics and Automation
RiverCentre, Saint Paul, Minnesota, USA
May 14-18, 2012

Loop Closure Through Vanishing Points in a Line-based Monocular SLAM

Guoxuan Zhang, Dong Hun Kang and Il Hong Suh, *Senior Member, IEEE*

Abstract— **In this paper, we present a vanishing point-based two-step loop closure method in a line-based monocular simultaneous localization and mapping (SLAM) system. Vanishing points can provide absolute directional landmarks for mobile robots. This guiding ability is important in that the observation of the vanishing points is invariant with respect to the robot pose. In our system, loop closure is performed in two steps: first, the accumulated heading error is reduced using an observation of previously registered vanishing points, and second, the observation of known floor lines allows for further pose correction. In this paper, we apply this method to solve the loop closure problem for a line-based SLAM within a corridor environment where the vanishing points are usually easily detectable. Experimental results show that our method is very efficient in a structured indoor environment.**

I. Introduction

Line-based visual simultaneous localization and mapping (SLAM), which uses lines as map features, has advantages over point-based SLAM methods. A line spans one-dimensional space; therefore, it is possible to construct a more expressive map within a structured environment using only lines. This implies that lines can produce a richer map than can be achieved using only points, even when using fewer features. The resulting map components of the line-based SLAM can easily be labeled with semantic names, which is a key element when robots have to communicate with or provide useful services to humans.

A 3D line can be modeled with two endpoints as in [1] and [2]. This is applicable in small workspace environments where the endpoints do not disappear from the camera view. Alternatively, the line entity can be modeled with an infinite line as in [3] and [4]. This is more suitable for long distance SLAM, and is not likely to be affected by occlusion or the disappearance of some line parts.

This paper extends the results of [5] and [6], incorporating vanishing points as a new map feature to address the unstable loop closure problem in previous work. Figure 1 shows the results of the above-mentioned work where the vertical and floor lines are used as feature data in a monocular indoor SLAM experiment. We adopted the vertical and floor lines as a linear feature in our system for two reasons: first, by using these two types of lines, it is still possible to represent many elements in the environment without the loss of important structural information. Second, except for the vertical and floor lines, the majority of the other features are lines which are positioned in such a way that, during the motion of the robot, it is hard to provide sufficient parallax to initialize these lines into mature features.

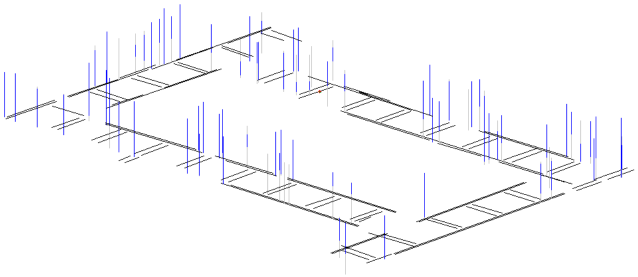


Fig. 1. The results of a line-based monocular SLAM. The vertical and floor lines were used as map features in this system.

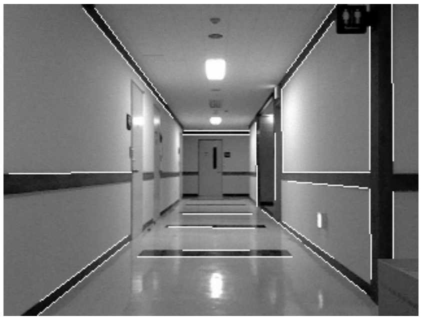


Fig. 2. Horizontal lines parallel to the robot's motion are hard to initialize in a monocular SLAM since the robot's movements provide less parallax for these line features.

For a 3D point feature, providing parallax means that the camera should deviate from the line connecting the point to the position from which the point feature was initially observed. On the other hand, for a 3D line feature, providing parallax means that the camera must deviate from the plane made by joining the line feature to the position of the initial observation. This implies that using lines as SLAM features has more constraints compared to using point features, and some lines are prone to remaining as immature features since the uncertainty of the line feature does not converge to a reliable threshold. For example, in Figure 2, the horizontal lines which are parallel to the motion of the robot are apt to be left as immature features. However, as we demonstrated in this paper, these lines can be used to extract vanishing points (VPs) to help the robot in correcting orientation errors.

This work was supported by the Global Frontier R&D Program on <Human-centered Interaction for Coexistence> funded by the National Research Foundation of Korea grant funded by the Korean Government(MEST) (NRF-M1AXA003-2011-0028353)

G. Zhang (imzgx@incorl.hanyang.ac.kr) and I. H. Suh (ihsuh@hanyang.ac.kr) are with the Department of Electronics and Computer Engineering, Hanyang University, Korea. All correspondence should be addressed to I. H. Suh.

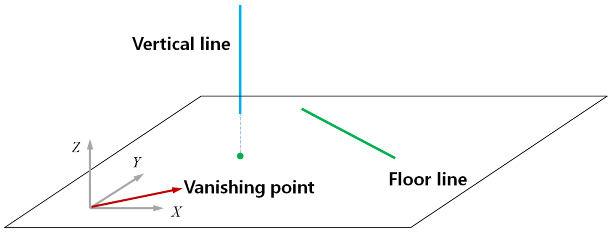


Fig. 3. The vertical line, floor line, and vanishing point together can be represented as 2D geometric objects. A vertical line is represented as a point, a floor line is represented as the same line, and a vanishing point is represented by a directional vector.

In an indoor environment, the robot's heading error is mostly generated from and accumulated at positions where the robot performs turning motions. Normally in such cases, there are few visual features that can be captured by the camera. The accumulated estimation error eventually degrades the quality of loop closure when the robot returns to a known position. This paper exploits the absolute guiding ability of VPs to alleviate the frequent problem of loop closure failure in our previously developed system. Since the VPs provide the direction information, it is independent of the current robot position and thus can help the robot to overcome the heading error.

In the field of robot navigation, vanishing points have been studied by many researchers. In [7] VPs are used as a means of clustering parallel lines with the same direction in the context of structure from motion study. In [8] VPs are generated via line features which are extracted from laser range finder readings. These then help the robot estimate its orientation and effectively initializes newly observed SLAM features. In [9] VP-based visual features are combined with a Hidden Markov Model to detect different corridor types when the robot approaches them.

The rest of this paper is structured as follows: Section II presents the parameterization method and observation model of both vertical and floor lines. In Section III, the visual processing and parameterization of VPs is introduced. Section IV describes the Extended Kalman Filter (EKF) using three different types of features in a line-based monocular SLAM. Experimental results using an actual robot platform are followed by our conclusions as presented in Sections V and VI, respectively.

II. The Vertical and Floor Line Features

A. The State Variables

In this paper the pose of the robot and the SLAM features are represented together on a 2D ground space. The robot performs planar motion on the floor, and features including vertical lines, floor lines, and VPs are represented as 2D entities as shown in Figure 3. Here, a floor line lies on the floor plane, a vertical line can be projected on the floor plane as a point, and a VP can simply be represented as a directional vector.

The SLAM system described in this paper is based on the EKF framework, where the state vector at time k is represented symbolically as:

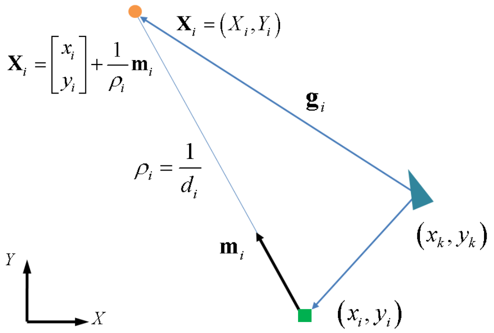

$\mathbf{X}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \frac{1}{\rho_i}\mathbf{m}_i$
$\mathbf{X}_i = (X_i, Y_i)$
$\mathbf{g}_i$
$\rho_i = \frac{1}{d_i}$
$\mathbf{m}_i$
(x_k, y_k)
(x_i, y_i)
Y
X

Fig. 4. A vertical line is represented as a point on the floor plane, and encoded using inverse depth parameterization.

$$\mathbf{x}_k = (\mathbf{r}_k, \mathbf{y})^T \tag{1}$$

where $\mathbf{r}_k$ and $\mathbf{y}$ denote the state of the robot and map features, respectively. The state of the robot is represented as:

$$\mathbf{r}_k = (x_k, y_k, \phi_k)^T \tag{2}$$

which encodes the translation and heading of the robot. The feature state vector $\mathbf{y}$ takes a different form for the vertical line, the floor line, and the VP according to different parameterization methods.

B. The Vertical Lines

As Figure 4 shows, a vertical line is projected on the floor plane as a 2D point $\mathbf{X}_i = (X_i, Y_i)^T$, and can therefore be encoded using inverse depth parameterization as:

$$\mathbf{y}_i = (x_i, y_i, \alpha_i, \rho_i)^T. \tag{3}$$

The vector $\mathbf{y}_i$ represents a ray connecting two points, $(x_i, y_i)^T$ and $\mathbf{X}_i$, where the point $(x_i, y_i)^T$ denotes the position where the vertical line was first observed, α_i is the angle between the ray and the X-axis, and ρ_i is the inverse distance of the ray, i.e. $\rho_i = 1/d_i$. Then, the position of a vertical line can be expressed using $\mathbf{y}_i$:

$$\begin{aligned} \mathbf{X}_i &= \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \frac{1}{\rho_i}\mathbf{m}_i && (4) \\ \mathbf{m}_i &= \begin{bmatrix} \cos\alpha_i \\ \sin\alpha_i \end{bmatrix} && (5) \end{aligned}$$

where $\mathbf{m}_i$ is the unit vector defining the direction of the ray. Despite a greater number of state variables, the ray-based inverse depth parameterization allows for the confinement of the main uncertainty of the feature into a single dominant dimension, while enabling the observation model to be high linear in the case of low parallax observation.

In Figure 4, the vertical line represented by $\mathbf{X}_i$ is observed by the robot in its current pose $(x_k, y_k)^T$ as a directional vector $\mathbf{g}_i^c = (g_x, g_y)^T$ in the camera frame:

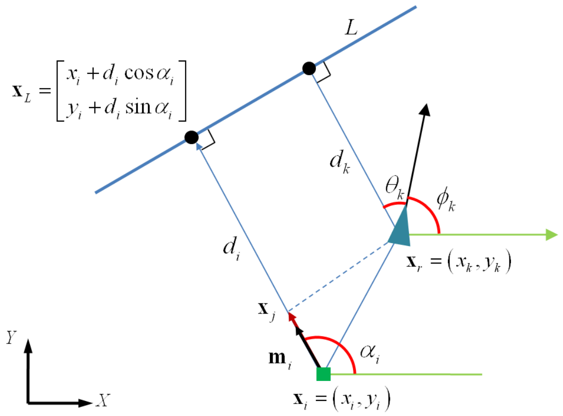

$\mathbf{x}_L = \begin{bmatrix} x_i + d_i\cos\alpha_i \\ y_i + d_i\sin\alpha_i \end{bmatrix}$
L
d_k
θ_k
ϕ_k
d_i
$\mathbf{x}_r = (x_k, y_k)$
$\mathbf{x}_j$
$\mathbf{m}_i$
α_i
$\mathbf{x}_i = (x_i, y_i)$
Y
X

Fig. 5. A floor line is represented as a line on the floor plane, and is encoded using the depth-range parameterization.

$$\mathbf{g}_i^c = \mathbf{R}^{CW}\left(\mathbf{X}_i - \begin{bmatrix} x_k \\ y_k \end{bmatrix}\right) \tag{6}$$

where $\mathbf{R}^{CW}$ denotes the rotation matrix from the world to the camera coordinate frame. Substituting equation (4) into (6) and normalizing the result by the inverse depth, ρ_i, leads to:

$$\tilde{\mathbf{g}}_i^c = \mathbf{R}^{CW}\left(\rho_i\left(\begin{bmatrix} x_i \\ y_i \end{bmatrix} - \begin{bmatrix} x_k \\ y_k \end{bmatrix}\right) + \mathbf{m}_i\right). \tag{7}$$

The normalized directional vector, $\tilde{\mathbf{g}}_i^c$, can be transferred to the horizontal offset, h_i, of the vertical line in the input image by applying the camera model:

$$h_i = c_x - f_x \frac{\tilde{g}_y}{\tilde{g}_x} \tag{8}$$

where c_x is the principal point, and f_x represents the focal length of the camera, both of which are given in the horizontal direction and represented by pixel values.

C. The Floor Lines

Figure 5 shows the feature representation of the floor line $\mathbf{L}_i$ on the floor plane. $\mathbf{x_L}$ is the incident point from the first observation point $\mathbf{x}_i = (x_i, y_i)^T$ to $\mathbf{L}_i$. x_i, y_i, α_i, d_i, and $\mathbf{m}_i$ retain the same meaning as they did in case of the vertical line feature. The floor line $\mathbf{L}_i$ is encoded as an incident point:

$$\mathbf{y}_i = (x_i, y_i, \alpha_i, d_i)^T. \tag{9}$$

However, unlike the vertical line feature, the depth d_i is now used instead of ρ_i.

We assume that the floor is flat, then the direct use of depth is possible because, for a fixed camera with a known height and angle to the floor plane, when an imaged line is back-projected to the floor plane, the transferred line can be observed with measurable depth and bearing. Even though the 4-vector representation is not a minimal encoding for the floor line constructed using range-bearing observation, local representation based on the camera frame can prevent the problem of increased feature uncertainty when the robot is far from the world origin. In Figure 5, if $\mathbf{x}_j$ is the incident point from the current robot pose $\mathbf{x}_r = (x_k, y_k)^T$ to the vector $\overrightarrow{\mathbf{x}_i\mathbf{x_L}}$, the observation model of the floor line can then be derived:

$$\overrightarrow{\mathbf{x}_i\mathbf{x}_j} = \mathbf{m}_i \ (\mathbf{m}_i \cdot \overrightarrow{\mathbf{x}_i\mathbf{x}_r}) \tag{10}$$

$$\overrightarrow{\mathbf{x}_j\mathbf{x_L}} = \overrightarrow{\mathbf{x}_i\mathbf{x_L}} - \overrightarrow{\mathbf{x}_i\mathbf{x}_j} = \begin{bmatrix} x_{jL} \\ y_{jL} \end{bmatrix} \tag{11}$$

$$\mathbf{h}_i = \begin{bmatrix} \theta_i \\ d_i \end{bmatrix} = \begin{bmatrix} \alpha_i - \phi_k \\ \sqrt{x_{jL}^2 + y_{jL}^2} \end{bmatrix}. \tag{12}$$

The sparseness and the bearing-distance encoding of the floor line decreases the possibility of false data association. This is the main reason that we adopted the floor line as the loop closure feature in the previous system.

III. The Vanishing Point Feature

A. Detection of The Vanishing Point

Parallel 3D lines viewed under perspective converge to a common point of intersection on the image plane known as the vanishing point (VP). The VP of a world line is obtained by intersecting the image plane with a ray parallel to the world line and passing through the camera center [10]. Thus, a VP depends only on the direction of the world line, and not on its position. This implies that the VP as a map feature does not have a fixed position on the map; it can only be defined by its directional information.

In indoor environments, several sets of parallel 3D lines may exist, and each set of parallel lines defines a common VP. Among these VPs, some can be imaged on the currently captured image, whereas others are not observable from the current viewpoint. In this paper, only VPs that have been observed at least once by the camera are added to the SLAM map.

The detection of VP is processed as follows: after extracting all lines from the current camera view, only diagonal lines are used to detect the VPs. As Figure 2 shows, a line group that is parallel to either a horizontal or a vertical line does not have a chance of converging to a visible VP in the current view. All crossing points made by two image lines are visible VPs candidates, and RANSAC was used to determine a single dominant VP in the current image. The initial value of the VP is taken as the average position of inter-crossing points of image lines. If the number of consistent image lines contributing to the dominant VP is less than a predefined threshold, the resulting VP is then considered to be weak and is discarded. The horizontal offset of the VP on the current image is fed to the next subsection and is used as the observation value of the VP feature.

B. Feature Representation

We derive the feature representation of a VP from the case of a vertical line feature. A vanishing point can be

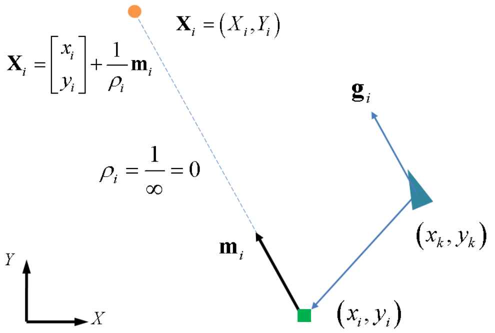
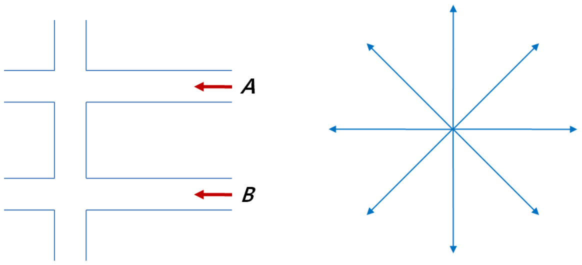

$\mathbf{X}_i = (X_i, Y_i)$
$\mathbf{X}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \frac{1}{\rho_i}\mathbf{m}_i$
$\mathbf{g}_i$
$\rho_i = \frac{1}{\infty} = 0$
(x_k, y_k)
$\mathbf{m}_i$
Y
X
(x_i, y_i)

Fig. 6. A vanishing point is represented as a directional vector in 2D space.



(a) The common observability of a VP

(b) The spacing of a VP

Fig. 7. The properties of a vanishing point on 2D ground space.

seen as an inverse depth point on the 2D ground space with infinite distance. By representing the VP with the notation $\mathbf{y}_i = (x_i, y_i, \alpha_i, \rho_i)^T$ as in the case of an inverse depth point, the 2D geometry of a VP is shown in Figure 6. This figure is much like Figure 4, except that the distance of the VP now becomes infinite. Furthermore, since the VP is located at infinity, the observation vector $\mathbf{g}_i$ is parallel to $\mathbf{m}_i$. Following Equations (4)–(6) and substituting the value $\rho = \frac{1}{\infty} = 0$, Equation (7) simply becomes:

$$\tilde{\mathbf{g}}_i^c = \mathbf{R}^{CW}\mathbf{m}_i. \tag{13}$$

This means that the observation of the VP is independent of robot translation, and is only dependent on the heading of the robot. Consequently, the inverse depth representation of the VP is reduced to:

$$\mathbf{y}_i = \alpha_i \tag{14}$$

where we can see that the variables x_i, y_i, ρ_i are independent of the representation of the VP, with the VP now being defined only by its direction. Thus, the observation model of Equation (8) can also be applied to the VP with the directional vector given by Equation (13).

For example, in Figure 7(a), the VPs observed from location A and B will have the same view angle value. Accordingly, they can be considered to be the same VP. It is clear that the 3D lines comprising the vanishing points A and B are all parallel in 3D space, and thus they will all share a common VP.

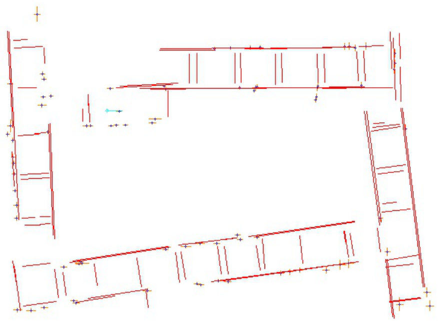


Fig. 8. An example of loop closure failure shown from top view. Accumulated heading error is the main cause for the failure in loop closure.

C. Loop Closure With the Vanishing Point

As already described in the Introduction, the accumulated heading error is the main cause of incorrect state estimation, and leads directly to the failure in loop closure. In our previous line-based SLAM system, loop closure is performed using only floor lines, where the floor line features are represented using homogeneous 2D objects on the ground plane. Here, *homogeneous* means that there are no distinctive descriptors used to discriminate each of the individual floor lines, but that they are differentiated by geometric information. The 3σ searching strategy is employed to conduct data-association, and to search for the loop closure feature when the robot revisits a known place. A loop closure method based on a homogeneous feature is prone to fail if the total accumulated heading error exceeds the 3σ searching boundary. Figure 8 shows an example of loop closure failure in this situation.

The VP has the ability to indicate absolute orientation. The performance is not affected by the robot translation, and therefore the VP is a good candidate for a SLAM landmark to alleviate the problem of loop closure failure. When the robot revisits a known place, observation of a previously registered VP can greatly decrease the heading error accumulated through the distant SLAM path, and prepares the robot for further loop closure using the floor lines. This two-step loop closure method has proven to be a very efficient and repeatable solution for the loop closure problem when applied on our line-based SLAM system.

Since the VP is only defined by its direction, it is a limited resource type when used as the map feature. This phenomenon is more apparent when the VP is defined on 2D space. We cannot excessively add different VPs to the same SLAM map because the robot heading error may confuse the process of VP-matching. As a result, even the mismatching of a single VP can induce catastrophic estimation failure. For the purpose of a successful loop closure in a SLAM trial, it is possible to analytically restrict the maximum uncertainty of the robot heading error when the total number of VPs is given. As in Figure 7(b), if the total number of VPs is set to 8, the angular spacing for a VP is then $\frac{\pi}{4}$. Under the 3σ searching strategy, the uncertainty of the robot heading should be controlled to be under the standard deviation of

$\frac{\pi}{24}$ to prevent incorrect data-association.

IV. The Line-based EKF-SLAM

Provided the observation model for the vertical line, the floor line and the VP, then the Jacobian of the observation over the state variables $\mathbf{H}$, the observational noise $\mathbf{R}$, and the innovation $\mathbf{v}$ for each of the line features and the vanishing points can be calculated. We use three sets of variables $\{\mathbf{H}_{VL}, \mathbf{R}_{VL}, \mathbf{v}_{VL}\}$, $\{\mathbf{H}_{FL}, \mathbf{R}_{FL}, \mathbf{v}_{FL}\}$, and $\{\mathbf{H}_{VP}, \mathbf{R}_{Vp}, \mathbf{v}_{VP}\}$ to denote these values. Subscripts $_{VL}$, $_{FL}$, and $_{VP}$ are the initials of the vertical line, the floor line, and the vanishing point, respectively. In our line-based SLAM system, the above variables are stacked as:

$$\mathbf{H}_L = \begin{bmatrix} \mathbf{H}_{VL} \\ \mathbf{H}_{FL} \\ \mathbf{H}_{VP} \end{bmatrix} \quad (15)$$

$$\mathbf{R}_L = \begin{bmatrix} \mathbf{R}_{VL} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{FL} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_{VP} \end{bmatrix} \quad (16)$$

$$\mathbf{v}_L = \begin{bmatrix} \mathbf{v}_{VL} \\ \mathbf{v}_{FL} \\ \mathbf{v}_{VP} \end{bmatrix} \quad (17)$$

where $\mathbf{H}_L$, $\mathbf{R}_L$, and $\mathbf{v}_L$ are the combined Jacobian of the observation model, the uncertainty of observation, and the innovation of the line-based SLAM system, respectively. Following the standard EKF algorithm [11], the measurement update can be completed using $\mathbf{H}_L$, $\mathbf{R}_L$, and $\mathbf{v}_L$.

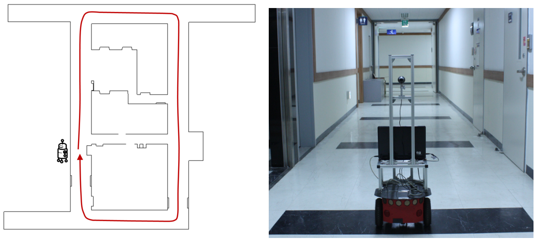
The vertical line, the floor line, and the VP are heterogeneous features because they have different feature representations, observation models, and a different number of observation variables. However, three features can inhabit the same EKF framework peacefully since the innovation covariance has normalized the heterogeneous properties between them, i.e. these features exhibit no difference in a single EKF computational system.

V. Experimental Results

A. Experiment

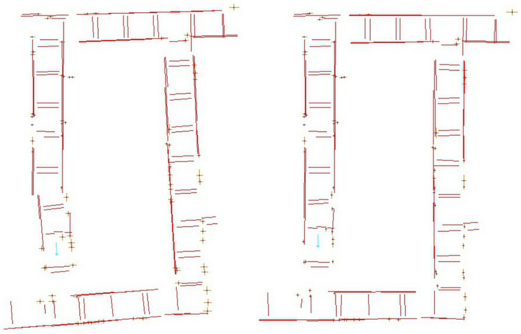
We tested our approach in actual robot experiments in a corridor environment, as shown in Figure 9 (a). The floor was flat and was decorated with rectangular black blocks. Each block had a width of $0.45m$, and they were spaced about $2.3m$ apart. A Pioneer 3-DX was used as the robot platform in the experiment, with a Logitech QuickCam E3500 webcam mounted on top as shown in Figure 9 (b). The camera was placed at a height of $100cm$ from the floor and faced forward. Images were collected at a resolution of 320×240 pixels by the camera at a control frequency of $3Hz$, and were recovered from the radial distortion. Before the experiment, the webcam was calibrated with a common checkboard method, and extracted intrinsic parameters were used in visual data processing.

In the feature initialization stage, image segments were extracted using a Hough transform and the endpoints were obtained by checking for pixel continuity. These extracted line segments were also used to detect the strong vanishing point feature using our algorithm described in Section III-A. As shown in Figure 9 (a), the robot started at the lower part of the left corridor, and traveled in a loop through the interior of the 6th floor of the IT Building at Hanyang University. The entire rectangular path has dimensions of about $11.5m \times 24m$, and the robot was manually driven during the experiment.



(a) Environment (b) Robot platform

Fig. 9. The experiment was conducted in a corridor environment and used a real robot platform.



(a) Before LCFL (b) After LCFL

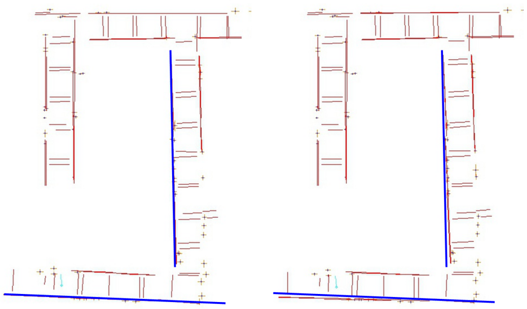
Fig. 10. The loop closure action of LCFL, where the loop closure is performed in one step, just before the map is completed.

B. Results

In the experiments, the vertical lines, the floor lines, and the vanishing points were used together as map features, and an EKF (using the algorithm described in the previous sections) was performed. We used both monocular images and odometer readings as input for our SLAM system. After the SLAM started, no special routines were dedicated to perform the loop closure process. Loop closure was automatically conducted as per the normal measurement update step of the SLAM algorithm.

We defined two different loop closure methods, loop closure with the floor line method (LCFL) is used in the previous system where only floor lines are used in loop closure; on the other hand, loop closure with the VP method (LCVP) is the method described in Section III-C, and now our current system use both LCFL and LCVP methods to perform loop closure.

We conducted two sets of experiments. The first experiment was devised to observe the different actions of the loop closure systems. In this experiment, the monocular images and odometer data of one single SLAM trial were saved



(a) Before LCVP (b) After LCVP

Fig. 11. The loop closure action of LCVP: this figure shows that the LCVP occurs earlier than the LCFL step; it occurred just after the robot turned at the fourth corner. The amount of map correction can be seen with the help of two auxiliary lines.

and used offline to compare the different actions between the previous system of using LCFL and the current system of using both LCVP and LCFL. Figure 10 depicts the loop closure action of our earlier system, where loop closure is performed in one step from (a) to (b). Most of the state estimation error is corrected in this step. In contrast, loop closure in the current system is performed in two steps: the step from (a) to (b) of Figure 11 corresponds to the LCVP step in which accumulated heading error is reduced by observing a known VP feature; and later, in the LCFL step, further loop closure is performed using the floor line feature. The algorithm used in the latter step is equivalent to the one in the previous system, but is now performed at a level of local refinement. The two steps of the current system provide a stable and safe mechanism to prevent the occurrence of loop closure failure as is verified in the experimental results presented next.

Table I shows the results of the second experiment which compared the performance of the two different loop closure systems. To enable this comparison, a total of 20 trials were performed, 10 for each of the two systems. The experiments were conducted under the same conditions as the first experiment. In Table I, the 'failures' indicate that the different map features are associated to the same one, and the resulting map is loop-closed in a topologically incorrect manner as depicted in Figure 8. For the LCFL-only system, 4 of 10 trials were failures due to accumulated heading error which prevented correct data-association. The results of the LCFL and LCVP hybrid method shows that additionally exploiting the VP as map features is more stable and has high repeatability in our experimental settings. No fails occurred in all of the 10 runs performed.

A demonstration video clip is available at [12]. The material offers important clues for a deeper understanding of the details in this paper. In the video, it can clearly be seen that the loop closure was performed in two steps: first, after the robot turned at the fourth corner, the heading error was corrected by observing the first VP; second, when the robot observed the floor line registered at the starting location, a finer loop closure was performed.

TABLE I

PERFORMANCE COMPARISON BETWEEN TWO LOOP CLOSURE METHODS

Methods	Results of loop closure (trials)	
	Successes	Failures
LCFL Method	6	4
LCVP + LCFL Method	10	0

VI. CONCLUSIONS AND FUTURE WORK

This paper proposes a vanishing point-based loop closure method in a line-based EKF-SLAM system. This method exploits the absolute directional guiding property of VPs to reduce the heading error whenever a known VP is observed by the robot. Loop closure is performed in a two-step procedure: the observation of VP corrects the heading error first, and a further loop closure follows when the floor line is observed. Experimental results indicate that this method is very efficient in an indoor corridor environment, with stable performance and sufficient repeatability. As one of our future works, we plan to include multiple vanishing points in a single update step to reliably upgrade the performance of line-based SLAM. We also plan to evaluate the performance of our proposed SLAM system in large scale structured environments including indoor and outdoor places.

REFERENCES

[1] P. Smith, I. Reid, and A. Davison, "Real-time monocular SLAM with straight lines," *in British Machine Vision Conference*, vol. 1, pp. 17-26, 2006.

[2] A. P. Gee, W. Mayol-Cuevas, Real-Time Model-Based SLAM Using Line Segments. 2nd International Symposium on Visual Computing. November 2006.

[3] T. Lemaire and S. Lacroix, "Monocular-vision based SLAM using line segments," *in Proc. of IEEE International Conference on Robotics and Automation*, Rome, Italy, pp. 2791-2796, 2007.

[4] J. Solà and T. Vidal-Calleja and M. Devy, "Undelayed initialization of line segments in monocular SLAM," *in Proc. of The IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 11-15, St. Louis, USA, 2009.

[5] G. Zhang, I. H. Suh, "SoF-SLAM: Segments-on-Floor-based Monocular SLAM," *in Proc. of The IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taiwan, 2010.

[6] G. Zhang, I. H. Suh, "Building a Partial 3D Line-based Map using a Monocular SLAM," *Accepted for IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011.

[7] M. Bosse, R. Rikoski, J. Leonard, and S. Teller, "Vanishing Points and 3D Lines from Omnidirectional Video," *in Proc. IEEE Intl Conf. Image Processing,* 2002.

[8] Y. H. Lee, C. Nam, K. Y. Lee, Y. S. Li, S. Y. Yeon, and N. L. Doh. "VPass: Algorithmic Compass using Vanishing Points in Indoor Environments," *in Proc. Of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2009.

[9] Y. B. Park, S. S. Kim and I. H. Suh, "Visual Recognition of Types of Structural Corridor Landmarks Using Vanishing Points Detection and Hidden Markov Models," *in Proc. of the International Conference on Pattern Recognition,* Istanbul, Turkey, 2010.

[10] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2nd Edition, pp. 213, 2004.

[11] H. Durrant-Whyte, T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99-110, Jun. 2006.

[12] http://youtu.be/HhsjbpJyacw