IEICE TRANSACTIONS

on Information and Systems

VOL. E101-D NO. 5
MAY 2018

A PUBLICATION OF THE INFORMATION AND SYSTEMS SOCIETY
The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER *Special Section on Machine Vision and its Applications*

Line-Based SLAM Using Non-Overlapping Cameras in an Urban Environment

Atsushi KAWASAKI[†], **Kosuke HARA**[††], ***Nonmembers***, *and* **Hideo SAITO**[†a)], ***Fellow***

SUMMARY We propose a method of line-based Simultaneous Localization and Mapping (SLAM) using non-overlapping multiple cameras for vehicles running in an urban environment. It uses corresponding line segments between images taken by different frames and different cameras. The contribution is a novel line segment matching algorithm by warping processing based on urban structures. This idea significantly improves the accuracy of line segment matching when viewing direction are very different, so that a number of correspondences between front-view and rear-view cameras can be found and the accuracy of SLAM can be improved. Additionally, to enhance the accuracy of SLAM we apply a geometrical constraint of urban area for initial estimation of 3D mapping of line segments and optimization by bundle adjustment. We can further improve the accuracy of SLAM by combining points and lines. The position error is stable within 1.5m for the entire image dataset evaluated in this paper. The estimation accuracy of our method is as high as that of ground truth captured by RTK-GPS. Our high accuracy SLAM algorithm can be apply for generating a road map represented by line segments. According to an evaluation of our generating map, true positive rate around the vehicle exceeding 70% is achieved.

key words: *SLAM, manhattan world, bundle adjustment*

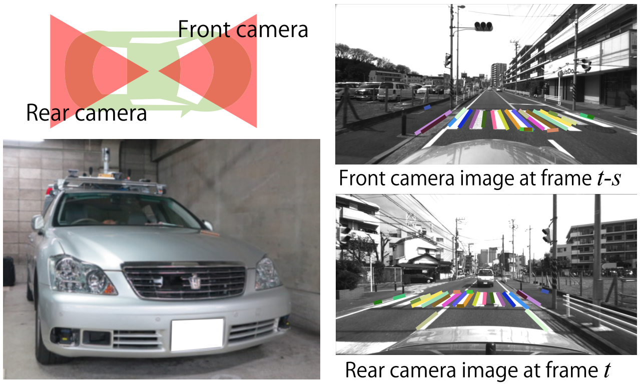

Front camera
Rear camera
Front camera image at frame *t-s*
Rear camera image at frame *t*

Fig. 1 Example of a multi-camera system on a vehicle and example of line segment matchings of different camera images. The line segments of the same color denote the correspondences.

1. Introduction

Advanced driver assistance system (ADAS) operate on the basis of the vehicle position, velocity, and traffic situation on the road. According to the technology report of a car manufacturer [1], the expected accuracy of the localization of vehicles for such systems is from a few dozen centimeters to a few meters. Achieving more accurate localization has therefore been an actively researched problem. High-end integrated accurate positioning system (e.g. POSLV [2]) achieves accuracy of up to several dozen centimeters by using an RTK-GPS receiver. However, this system is not suitable for automotive systems because its cost is extremely high and its accuracy depends on the reception of microwaves and radio waves from satellites. Alternately, visual SLAM is drawing the attention of a lot of researchers. State-of-the-art visual SLAM can achieve the same accuracy as a laser range scanner when the scene has stable feature points. However, if few feature points are detected from the scene that mainly consist of texture-less surfaces, visual SLAM usually suffers a large position and angle error. Some systems use line features or a hybrid of points and lines for SLAM.

By using multiple cameras, the accuracy of visual SLAM can generally be improved. Recent vehicles often have cameras that capture the front and rear views for safety. We aim to improve the accuracy of SLAM by finding corresponding line segments between different camera images as shown in Fig. 1. These correspondences are referred to as “inter-camera correspondences”. In contrast, corresponding line segments between different frames captured by the same camera are referred to as “intra-camera correspondences”. Inter-camera correspondences can be collected by moving the vehicle forward, so that the rear camera can capture the area that was previously captured by the front camera. Since such correspondences have a wide baseline between the cameras, more accurate localization and mapping is expected. In our previous work [3], we proposed a method for improving the accuracy of motion estimation by finding the inter-camera corresponding points. It used a novel feature-point-matching algorithm that warps feature patches on the basis of Manhattan World assumption [4].

In this paper, we propose a line-based SLAM, in which our warping algorithm is applied to line segments matching. Furthermore, the geometrical constraint of an urban area is incorporated into the initial estimation of 3D line segments and the optimization by bundle adjustment. The accuracy of the estimation is further improved by combining the proposed line-based SLAM with point-based SLAM [3].

The proposed line-based SLAM is applied to generate a digital map. Digital maps are generally used for purposes

Manuscript received August 24, 2017.
Manuscript revised December 4, 2017.
Manuscript publicized February 16, 2018.

[†] The authors are with the Department of Information and Computer Science, Keio University, Yokohama-shi, 223–8522 Japan.

[††] The author is with Denso IT Laboratory, Tokyo, 150–0002 Japan.

a) E-mail: hs@keio.jp

DOI: 10.1587/transinf.2017MVP0006

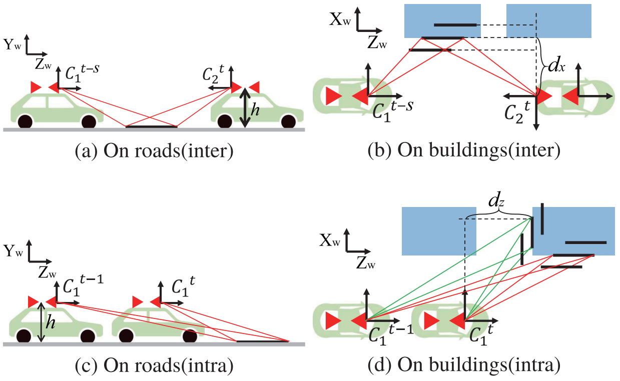
such as navigation, self driving, and ADAS. Among the various kinds of digital maps, road maps represented by line segments and spline curves with information such as white lines, road markings, and curbs have recently been drawing attention. The recently developed mobile mapping system (MMS) [5] makes it possible to generate these maps. As examples of works using maps, Schreiber et al. [6] proposed a method of vehicle localization based on cross-checking detected line segments against the map. These digital maps have to be accurate and up to date, because traffic information is frequently updated. The conventional method of generating accurate maps needs a survey vehicle equipped with RTK-GPS, multiple cameras, and laser scanners. However, it is not realistic to use a survey vehicle every time the road information is updated. To reduce time and effort of using a survey vehicle, we aim to generate line-based road map generation, automatically.

2. Related Works

To improve the accuracy of SLAM, multi-camera based SLAM has recently been developed. For example, Kazik et al. [7] presented a multi-camera system that performs absolute scale motion. However they noted that degenerate cases occur if both cameras move in straight lines. In [8], [9], cameras motion, including scale, can be recovered in the degenerate straight motion case. Lee et al. [8] combined the known yaw angle to one inter-camera correspondence to retrieve the scale. Pless [10] determined the relationship between the accuracy and different camera designs. He showed that the best design for a two camera system is to place cameras facing in opposite directions with their optical axes aligned. Our camera design is same as them.

Line-based SLAM is also drawing attention. Elqursh et al. [11] presented a method for estimating a relative camera pose between two images from lines. Their method requires only three lines, with two of which are parallel and orthogonal to the third. Smith et al. [12] demonstrated a real-time line-based SLAM that extended the point-based Extended Kalman Filter (EKF) SLAM system to line correspondences. They detected lines by checking if Sobel-detected edges exist between two corners or not. Hirose et al. [13] proposed a novel descriptor of line segment features for line-based SLAM. Many other researches on SLAM combine lines and other features. Koletschka et al. [14] proposed a method of motion estimation using points and lines by stereo line matching. Lu et al. [15] incorporated a combination of points, lines, planes, and vanishing points into bundle adjustment. Zhou et al. [16] proposed a novel visual SLAM (using EKF) based on Manhattan World assumption using points and lines. Manhattan World assumption, proposed Coughlan et al. [4] in 1999, states that most planar surfaces in urban scenes lie in one of three mutually orthogonal orientations. Applying this assumption, Furukawa et al. [17] modified 3D model generated by multi-view stereo.

Inspired by the previous works, we aim to improve the accuracy of SLAM for a vehicle driving in an urban area. To find corresponding line segments captured by non-overlapping cameras, we propose a novel matching algorithm based on Manhattan World assumption. Additionally, a geometrical constraint of an urban city area is applied for initial estimation of 3D mapping of line segments and optimization by bundle adjustment. The proposed SLAM method also generates a road map consisting of line segments that can be utilized by ADAS.



(a) On roads(inter) (b) On buildings(inter)

(c) On roads(intra) (d) On buildings(intra)

Fig. 2 Examples of inter-camera and intra-camera correspondence.

3. System Overview

A line-based SLAM pipeline is overviewed as follows. The proposed line-based SLAM system takes a wheel odometry and two consecutive frames from the front and rear cameras as an input. Odometry refers to the use of data from wheel sensors to estimate a relative position over time. For each frame t, the system detects line segments from camera images using Line Segment Detector (LSD) algorithm [18]. It then warps the patches around a line segment to match them in the following three cases. The first case is matching between the front image features at frame t and $t-1$; the second case is between the rear image features at frame t and $t-1$; and the third case is between the rear image at frame t and the front image at frame $t-s$. By finding the best match for various s values, we need to find correspondences between a rear image and multiple front images. Example of matchings are shown in Fig. 2. Each line segment is tracked over different frames as long as corresponding line segments can be found in later frames. After 3D mapping of line segments are initially estimated by using those correspondences, bundle adjustment is applied to the 3D lines with the initial translation information measured by the odometry.

4. Notation

4.1 Projection Model of Cameras

In this section, notation for representing projection model of the cameras in the proposed method is presented as indicated in Eq. (1)-(7). This is used for computing the reprojection error in the bundle adjustment in the proposed method. To create a multi-camera system, it is necessary to define four coordinate systems W, ${C_1}^t$, ${C_2}^t$ and V^t which correspond to world, front camera, rear camera and vehicle coordinate systems at frame t, respectively. The rotation and position of a vehicle at frame t is expressed with ${\mathbf{R}_{\mathrm{vw}}}^t$, ${\mathbf{T}_{\mathrm{vw}}}^t$. The relative transformations from the vehicle coordinate system to the front or the rear camera coordinate system are expressed with $\mathbf{R}_{\mathrm{c_1v}}$, $\mathbf{T}_{\mathrm{c_1v}}$, $\mathbf{R}_{\mathrm{c_2v}}$, $\mathbf{T}_{\mathrm{c_1v}}$. These transformations are determined by the relative pose and position of the front and rear cameras which are calibrated in advance. A point $\mathbf{p}$ on world coordinate system W is mapped to the camera frame coordinates and then to the normalized image plane as follows:

$$(u_1 \;\; v_1)^{\mathrm{T}} = \pi\left(\mathbf{R}_{\mathrm{c_1v}}\left({\mathbf{R}_{\mathrm{vw}}}^t\mathbf{p} + {\mathbf{T}_{\mathrm{vw}}}^t\right) + \mathbf{T}_{\mathrm{c_1v}}\right) \tag{1}$$

$$(u_2 \;\; v_2)^{\mathrm{T}} = \pi\left(\mathbf{R}_{\mathrm{c_2v}}\left({\mathbf{R}_{\mathrm{vw}}}^t\mathbf{p} + {\mathbf{T}_{\mathrm{vw}}}^t\right) + \mathbf{T}_{\mathrm{c_2v}}\right) \tag{2}$$

$$\pi\begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} = \begin{pmatrix} q_x/q_z \\ q_y/q_z \end{pmatrix} \tag{3}$$

The 3D line $\mathbf{L}$ is given as 6-vector $(\mathbf{p}^{\mathrm{T}}, \mathbf{r}^{\mathrm{T}})^{\mathrm{T}}$. The 3-vectors $\mathbf{p}$ and $\mathbf{r}$ are a point on the 3D line and the direction of the 3D line, respectively. Point $\mathbf{u}$ and direction $\mathbf{a}$ in the normalized image plane are computed as follows:

$$\mathbf{q}_k = \begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} = \mathbf{R}_{\mathrm{c_kv}}({\mathbf{R}_{\mathrm{vw}}}^t\mathbf{p} + {\mathbf{T}_{\mathrm{vw}}}^t) + \mathbf{T}_{\mathrm{c_kv}} \tag{4}$$

$$\mathbf{A}_k = \begin{pmatrix} A_x \\ A_y \\ A_z \end{pmatrix} = \mathbf{R}_{\mathrm{c_kv}}{\mathbf{R}_{\mathrm{vw}}}^t\mathbf{r} \tag{5}$$

$$\mathbf{u}_k = \pi(\mathbf{q}_k) \tag{6}$$

$$\mathbf{a}_k = \pi(\mathbf{q}_k + \mathbf{A}_k) - \pi(\mathbf{q}_k) \simeq \begin{pmatrix} \mathrm{q}_z\mathrm{A}_x - \mathrm{q}_x\mathrm{A}_z \\ \mathrm{q}_z\mathrm{A}_y - \mathrm{q}_y\mathrm{A}_z \end{pmatrix} \tag{7}$$

where k denotes camera 1 or camera 2. If $k = 1$, the 3D line $\mathbf{L}$ is observed by the front camera. The 2D line $\mathbf{l}_k$ is given as 4-vector $({\mathbf{u}_k}^{\mathrm{T}}, {\mathbf{a}_k}^{\mathrm{T}})^{\mathrm{T}}$.

4.2 Wheel Odometry

In this section, we present notation of geometry related to the wheel odometry used as the intial value of the bundel adjustment and computing the warp function for finding correspondences between the cameras.

We define the wheel odometry $\mathbf{x}$ as:

$$\mathbf{x}^{t+1} = \begin{pmatrix} x^{t+1} \\ z^{t+1} \\ \theta^{t+1} \end{pmatrix} = \mathbf{x}^t + \Delta\mathbf{x}^t + {\varepsilon_{\mathbf{x}}}^t \tag{8}$$

As for the proposed SLAM algorithm, 6-DOF (Degree of Freedom) motion is estimated. However, the odomery is 3-DOF due to characteristics of our application. $\Delta\mathbf{x}^t$ is predicted relative movement from a previous time. $\mathbf{x}^0$ is the initial position of the vehicle. When $\mathbf{x}^0$ coincides with the origin of the world coordinate system, $\mathbf{x}^t$ is another way of expressing ${\mathbf{R}_{\mathrm{vw}}}^t$ and ${\mathbf{T}_{\mathrm{vw}}}^t$. And ${\varepsilon_{\mathbf{x}}}^t$ is the noise of $\Delta\mathbf{x}^t$, which is assumed to be zero-mean Gaussian white noise with covariance $\Sigma_{\mathbf{x}}$.

$${\varepsilon_{\mathbf{x}}}^t \sim \mathcal{N}(0, \mathbf{\Sigma}_{\mathbf{x}}) \tag{9}$$

$$\mathbf{\Sigma}_{\mathbf{x}} = \begin{pmatrix} {\sigma_x}^2 & 0 & 0 \\ 0 & {\sigma_z}^2 & 0 \\ 0 & 0 & {\sigma_\theta}^2 \end{pmatrix}\Delta t \tag{10}$$

where σ_x, σ_z, and σ_θ are error variances. Assuming that ${\varepsilon_{\mathbf{x}}}^t$ simply increases in proportion to time, the covariance increases in proportion to the time intervals between image acquisitions, Δt.

$\Delta\mathbf{x}^t$ is estimated by wheel speed in the prediction step of EKF. It can be computed by using the same formula as the geometric model of the two-wheeled robot [19]. We define the velocity of the center of the rear wheels and the angular velocity as v_t and ω_t, respectively. These are computed as follows:

$$v_t = \frac{({v_t}^L + {v_t}^R)}{2}, \omega_t = \frac{({v_t}^R - {v_t}^L)}{2d_a} \tag{11}$$

where d_a is the installation interval between the rear wheels. Assuming that the vehicle is circularly moved in a radius ρ and the vehicle direction is changed to ω_t for Δt, a radius ρ is v_t/ω_t. $\Delta\mathbf{x}^t$ is described as follows.

$$\begin{aligned} \Delta\mathbf{x} &= \begin{pmatrix} \rho(-\cos(\theta_t + \omega_t\Delta t) + \cos\theta_t) \\ \rho(\sin(\theta_t + \omega_t\Delta t) - \sin\theta_t) \\ \omega_t\Delta t \end{pmatrix} \\ &= \begin{pmatrix} v_t\Delta t\,\mathrm{sinc}(\frac{\omega_t\Delta t}{2})\sin(\theta_t + \frac{\omega_t\Delta t}{2}) \\ v_t\Delta t\,\mathrm{sinc}(\frac{\omega_t\Delta t}{2})\cos(\theta_t + \frac{\omega_t\Delta t}{2}) \\ \omega_t\Delta t \end{pmatrix} \end{aligned} \tag{12}$$

5. Proposed Line Matching Method

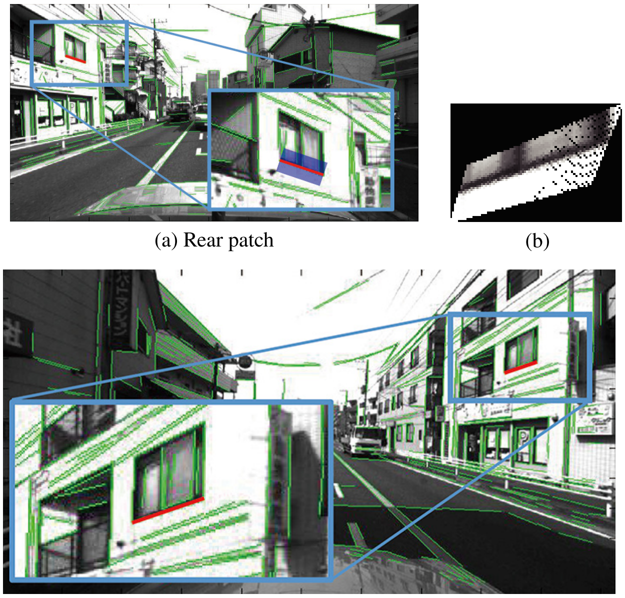
5.1 Inter-Camera Correspondences

First, we focus on the method of getting the correspondences between the front and the rear images.

As shown in Fig. 3-(a), we define a patch as a rectangle having a pre-defined width (20 pixels in this paper, experimently decided) in the normal direction centered on the detected red line segment. For finding the corresponding line segment in the front image by template matching with the patch of the rear image, we need to make the perspective appearance of the patch of the rear image similar to the front image by warping the patch as shown in Fig. 3-(b). The warping function can be derived by 3D surface structure of the patch region, but it is actually not easy to be obtained just from just from the images.

Here, we assume that the 3D surface of the patch region is either a part of the road or wall of buildings, which can be considered as perpendicular or paralell to the travel direction(z-axis) of the camera according to Manhattan World Assumption. Based on this assumption, we can derive the 3D coordinate for all pixels in the patch region as follows.

By using the center point $\mathbf{u}_{m,2}$ of the line segment in the



(a) Rear patch

(b)

(c) Front patch

Fig. 3 (a) is a rear image, where the blue rectangle denotes the original patch of the red detected line, (b) is the warped patch transformed from (a), and (c) is a front image. Warped patch (b) is similar to the surroundings of the red line in (c).

rear normalized image, all pixels of the patch in the normalized image plane are expressed as $\mathbf{u}_2 = (u_{m,2} + \alpha, v_{m,2} + \beta)^{\mathrm{T}}$, where α and β the displacement from the center point.

In the case of matching between front and rear, we need to consider two cases of the 3D surface of the patch: one is on the road, and the other is on the front wall of buildings as illustrated in Fig. 2-(a) and (b).

When 3D surface of the patch region is on the road surface, which is flat and parallel to the travel direction(z-axis), all y-coordinate value of the 3D point in the patch in ${C_2}^t$ should have the camera installation height h. Since y-coordinate value in the image for the patch is $(v_{m,2} + \beta)$, all coordinate values of the 3D points ${\mathbf{p}_\mathrm{r}}^t$ in the patch are derived by multiplying with $h/(v_{m,2} + \beta)$ as follows:

$$
{\mathbf{p}_\mathrm{r}}^t = h/(v_{m,2} + \beta)\begin{pmatrix}\mathbf{u}_2^t \\ 1\end{pmatrix} = \begin{pmatrix} h(u_{m,2} + \alpha)/(v_{m,2} + \beta) \\ h \\ h/(v_{m,2} + \beta) \end{pmatrix} \qquad (13)
$$

where $\mathbf{u}_2^t$ indicates a 2D position of the rear normalized coordinates.

In another case that the 3D surface of the patch region is on the front wall of the buildings, all x-coordinate values of the 3D points in the patch in ${C_2}^t$ take value d_x, distance to the building on the x-axis in ${C_2}^t$ in Fig. 2-(b). Since x-coordinate value in the image for the patch is $(u_{m,2} + \alpha)$, all coordinate values of the 3D point ${\mathbf{p}_\mathrm{bf}}^t$ in the patch are derived by multiplying with $d_x/(u_{m,2} + \alpha)$ as follows:

$$
{\mathbf{p}_\mathrm{bf}}^t = d_x/(u_{m,2} + \alpha)\begin{pmatrix}\mathbf{u}_2^t \\ 1\end{pmatrix} = \begin{pmatrix} d_x \\ d_x(v_{m,2} + \beta)/(u_{m,2} + \alpha) \\ d_x/(u_{m,2} + \alpha) \end{pmatrix} \qquad (14)
$$

where $\mathbf{u}_2^t$ indicates a 2D position of the rear normalized coordinates.

By mapping the 3D positions in the patch for both cases in ${C_2}^t$ to the front-image coordinate in ${C_1}^{t-s}$, we can derive the following geometric transformations for warping the rear-image patch as shown in Fig. 3-(b).

$$
{\mathbf{u}_1}^{t-s} = \pi(\mathbf{R}_{\mathrm{c_1 v}}({\mathbf{R}_{\mathrm{vw}}}^{t-s}({\mathbf{R}_{\mathrm{wc_2}}}^t{\mathbf{p}_\mathrm{r}}^t + {\mathbf{T}_{\mathrm{wc_2}}}^t) + {\mathbf{T}_{\mathrm{vw}}}^{t-s}) + \mathbf{T}_{\mathrm{c_1 v}}) \qquad (15)
$$

$$
\mathbf{u}_1^{t-s} = \pi(\mathbf{R}_{\mathrm{c_1 v}}({\mathbf{R}_{\mathrm{vw}}}^{t-s}({\mathbf{R}_{\mathrm{wc_2}}}^t{\mathbf{p}_\mathrm{bf}}^t + {\mathbf{T}_{\mathrm{wc_2}}}^t) + {\mathbf{T}_{\mathrm{vw}}}^{t-s}) + \mathbf{T}_{\mathrm{c_1 v}}) \qquad (16)
$$

where $\mathbf{p}_\mathrm{r}$ and $\mathbf{p}_\mathrm{bf}$ are 3D positions of the patch for both cases in ${C_2}^t$. These equations map the position of the rear normalized image coordinate ${\mathbf{u}_2}^t$ at frame t to the front normalized image coordinate ${\mathbf{u}_1}^{t-s}$ at frame $t - s$. The patch can be warped by applying these transformations to all pixels contained in the patch.

Subsequently, the warped patch and the feature patch in the front image are matched. Even if the patch is accurately warped, the projected line does not exactly match the line segment owing to the error of the odometry. Therefore, an error ellipse based on EKF proposed by Davison et al. [20] is considered. The error ellipse can be drawn using covariance $\mathbf{\Sigma}_F$ as follows:

$$
\mathbf{\Sigma}_F = \frac{\partial \mathbf{w}}{\partial \mathbf{x}}(s\mathbf{\Sigma}_\mathbf{x})\frac{\partial \mathbf{w}}{\partial \mathbf{x}}^{\mathrm{T}} \qquad (17)
$$

In our previous method [3], we superposed the warped patch on feature points within the error ellipse and compared them. However, in the case of line segment matching, it is difficult to superpose the warped patch on detected line segments because the positions of the endpoints of line segments detected by LSD are always ambiguous. Therefore, a zero-mean normalized cross-correlation (ZNCC) score is computed by raster scan of the warped patch within the error ellipse. On the position that the ZNCC score is the highest, we compute the angle between the warped line segment and the detected line segments in the front image and the distances from the endpoints of detected line segments to the warped line segment. The line segment which have smaller length and angle than the threshold and the smallest is adopted as the correspondence.

As described above, we presented a way of computing the matching score, but two undecided elements still exist. One is that we do not know which line segment exists on buildings or roads in the detection stage. The other is that the correct value of d_x is unknown when the line segment exists on the front wall of buildings. To solve these undecided elements, it is necessary to explore all possibilities per detected line segment in the rear image. After trying Eqs. (8) and (9) and changing d_x at regular intervals (0.5m) from 0m to 20m, the line segment that gets the highest ZNCC score is taken as the correspondence.

5.2 Intra-Camera Correspondences

Our method can be applied for matching line segments between consecutive frame pairs (frame t and $t - 1$) of front

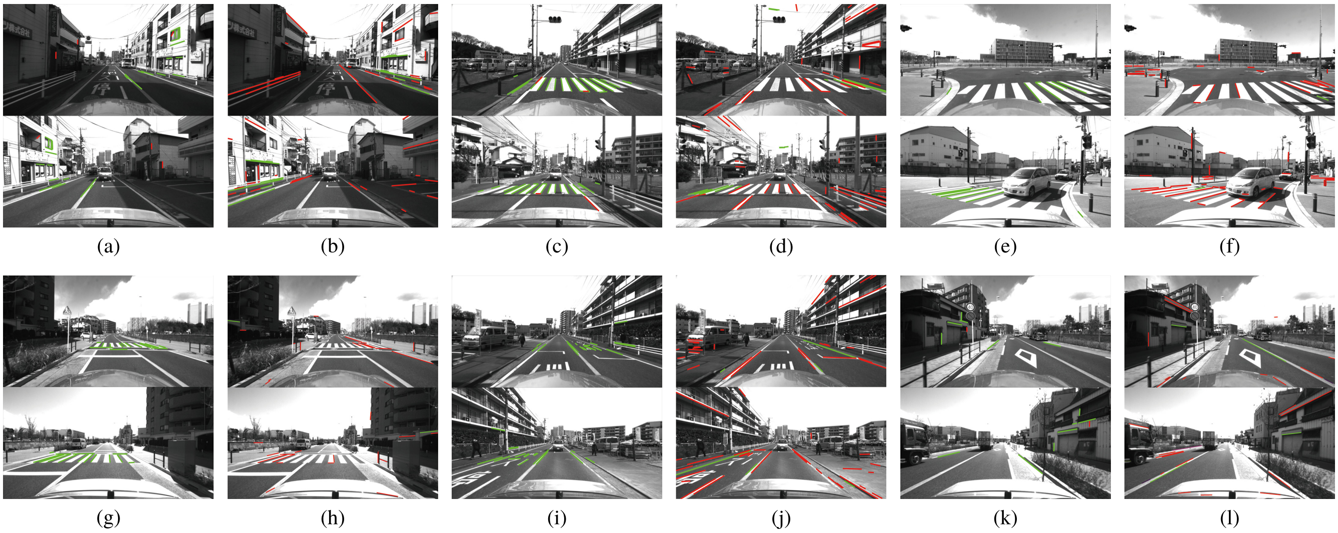


Fig. 4 Examples of matching by our method (a, c, e, g, i, k) and LEHF [13] (b, d, f, h, j, l). These images are matched pairs of front images and rear images. Green lines and red lines shows correct and wrong matching, respectively. Pairs of (a, b), (c, d), . . . , (k, l) are same images and show comparison of our method and LEHF.

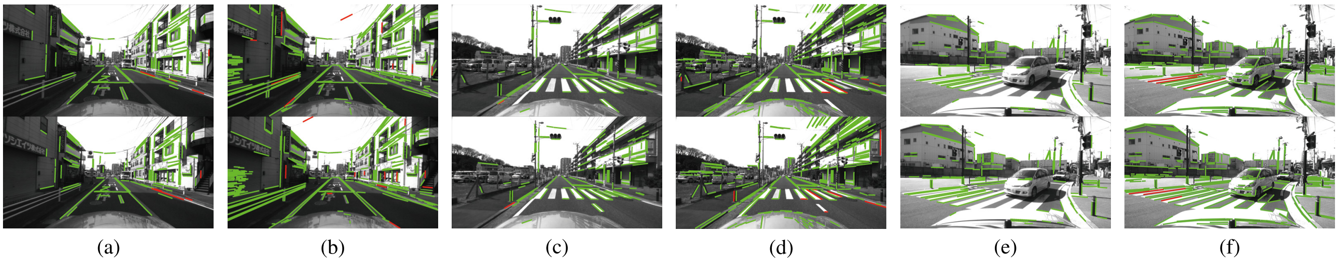


Fig. 5 Examples of matching by our method (a, c, e) and LEHF [13] (b, d, f). (a-d) are matched pairs of front images. (e, f) are matched pairs of rear images. Pairs of (a, b), (c, d), and (e, f) are same images and show comparison of our method and LEHF.

images or pairs of rear images. In these matching, we can consider three cases of the 3D surface of the patch. Two cases are the same as inter-camera correspondences. The last is in the case that the 3D surface of the patch is on the side wall of buildings as illustrated in Fig. 2-(c) and (d). In this case, the 3D surface around the line segment is flat and vertical to the travel direction (z-axis) in accordance with Manhattan World assumption. Since all z-coordinates of the 3D points in the patch have the same value d_z which is the distance from the vehicle to the side wall of the building on the z-axis, all coordinate values of the 3D point in the patch are derived by multiplying by d_z as follows:

$$\mathbf{p}_{\mathrm{bs}} = \mathrm{d}_{\mathrm{z}} \begin{pmatrix} \mathbf{u}_2^t \\ 1 \end{pmatrix} = \begin{pmatrix} d_z(u_{m,1} + \alpha) \\ d_z(v_{m,1} + \beta) \\ d_z \end{pmatrix} \tag{18}$$

By mapping the 3D positions in the patch in ${C_2}^t$ to the front-image coordinate in ${C_1}^{t-s}$, we can derive the following geometric transformations for warping.

$$\mathbf{u}_1{}^{t-1} = \pi(\mathbf{R}_{\mathrm{c}_1\mathrm{v}}(\mathbf{R}_{\mathrm{vw}}{}^{t-1}(\mathbf{R}_{\mathrm{wc}_1}{}^{t}\mathbf{p}_{\mathrm{bs}} + \mathbf{T}_{\mathrm{wc}_1}{}^{t}) + \mathbf{T}_{\mathrm{vw}}{}^{t-1}) + \mathbf{T}_{\mathrm{c}_1\mathrm{v}}) \tag{19}$$

It maps the position of the front (rear) normalized image coordinate $\mathbf{u}_1{}^t$ at frame t to the front (rear) normalized image coordinate $\mathbf{u}_1{}^{t-1}$ at frame $t-1$.

The method of computing the matching score is the same as the method when finding the inter-camera correspondences. In this matching, we do not know d_z and need to change d_z at regular interval (0.5m) from 0m to 20m.

5.3 The Matching Result

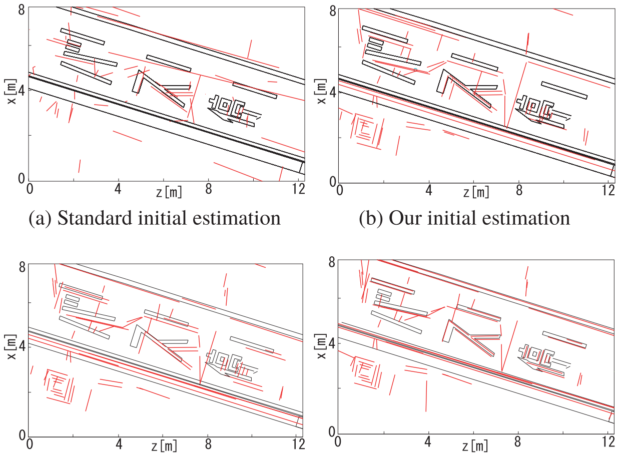
The results of matching between front and rear images are shown in Fig. 4. Figure 4-(a, c, e, g, i, k) are obtained by our method, meanwhile Fig. 4-(b, d, f, h, j, l) are by LEHF [13]. It is clear that our method can not detect many corresponding line segments, but it achieves higher matching accuracy than that of LEHF. The results of matching of pairs of front images and pairs of rear images are shown in Fig. 5. Intra-camera correspondences by our method (Fig. 5-(a, c, e)) are as accurate as those given by LEHF (Fig. 4-(b, d, f)). The advantage of our method is that it can classify all matched line segments into “on the building walls” or “on the road”. This information is helpful for the next step. The line classification is failed when Manhattan world assumption is not established, for example, the vehicle makes a turn. In that case, however, we can get the inter-camera correspondences on the road and the intra-camera correspondences because intra-camera correspondences do not have large differences of viewing direction.

6. Initialization and Optimization

6.1 Initial Estimation of 3D Line

As for line-based SLAM, 3D lines must be computed from correspondences. Although various method for computing 3D lines are available, the most popular one is based on line of intersection of planes which pass the camera center and the line segment. However, this method is not suitable for vehicle SLAM, because a 3D line cannot accurately be computed especially when the angle between planes passing the line segments parallel to the travel direction is small. As shown in Fig. 6-(a), most 3D line segments are pointing in imprecise directions.

Under the assumption that 3D lines exist on three dominant planes, accurate 3D lines are computed. As explained at the end of Sect. 5, under Manhattan World assumption, the matched line segments are classified as existing on the front wall of buildings, the side wall of buildings, or the road surface. In the first two cases, we got the distances d_x or d_z between the building walls and the vehicle in the process of line segment matching. According to the 3D geometrical information of the matched line segments, 3D line segments can be computed by projecting all matched line segments onto the plane of the front wall of buildings ($x = d_x$), the plane of the side wall of buildings ($z = d_z$), or the road surface ($y = h$), respectively. By selecting 3D line segments with smaller re-projection errors, it is possible to collect 3D line segments as initial estimation of 3D mapping. The re-projection error is computed from the perpendicular distance from a re-projection of 3D line to the endpoints of a detected line segment in the image plane. This method computing the re-projection error of line segments is widely used and defined in [21]. The 3D line segments determined by our initial estimation method are shown in Fig. 6-(b).



(a) Standard initial estimation

(b) Our initial estimation

(c) Without the assumption

(d) With the assumption

Fig. 6 Examples of road maps: (a) is the initial estimation by line intersection, (b) is one by our method, and (c) and (d) are the results after bundle adjustment without and with the assumption.

6.2 Optimization by Bundle Adjustment

In this step, we focus on optimization of 3D lines and car positions. The optimization is performed every time when the newest frame is obtained. The set of corresponding line segments Ω is defined as:

$$\begin{aligned}\Omega =&\{\omega_i = (t, k, j)| \\ &t \in \{1, \dots, T\}, k \in \{1, 2\}, j \in \{1, \dots, J\}\}\end{aligned} \tag{20}$$

where the ith line segment denotes that a 3D line $\mathbf{L}^j$ is observed by camera k at frame t. 2D line $\mathbf{l}_i$ denotes the re-projection of $\mathbf{L}^j$ in camera k image. We use bundle adjustment to optimize both the pose (${\mathbf{R}_{\mathrm{vw}}}^t$, ${\mathbf{T}_{\mathrm{vw}}}^t$) and 3D lines through re-projection error minimization. The endpoints of observed line segment are expressed with ${\mathbf{g}_1}^i$ and ${\mathbf{g}_2}^i$. The cost function is define as:

$$E = \sum_i \sum_{n=1}^{2} ({e_n}^i)^2 = \sum_i \sum_{n=1}^{2} d^2{}_\perp({\mathbf{g}_n}^i, \mathbf{l}^i) \tag{21}$$

where $d_\perp(\cdot, \cdot)$ denotes the perpendicular distance from a point to a line in images. We find ${\mathbf{R}_{\mathrm{vw}}}^t$, ${\mathbf{T}_{\mathrm{vw}}}^t$, and $\mathbf{L}^j$ which minimize E. The initial values of ${\mathbf{R}_{\mathrm{vw}}}^t$, ${\mathbf{T}_{\mathrm{vw}}}^t$ correspond with the odometry. Bundle adjustment is conducted by using the iterative non-linear Levenberg-Marquardt optimization algorithm with numerical differentiation based on [22].

To improve the accuracy of the optimization, a constraint is incorporated into the cost function. Under the assumption that road surface is flat, the constraint such that 3D lines labeled as "on the road" only moves on plane $y = h$ is incorporated into bundle adjustment. A similar constraint for 3D lines labeled as "on the building" is not incorporated because the positions of the planes of the building walls, represented by d_x and d_z, are just roughly estimated by the patch matching with regular interval (0.5m) as explained in Sec.5.

The above-mentioned bundle adjustment under the planar constraint of the road is described in detail as follows. Two types of Jacobian matrices for poses and 3D lines should be used in solving nonlinear least square problem by Levenberg-Marquardt method. Among them, we focus on the Jacobian matrix for 3D lines.

The Jacobian for 3D lines needs partial differentials with respect to point $\mathbf{p}$ and direction $\mathbf{r}$. According to chain rule, the Jacobian matrices can be derived as:

$$\frac{\partial {e_n}^i}{\partial \mathbf{p}^j} = \frac{\partial {e_n}^i}{\partial \mathbf{q}^i} \frac{\partial \mathbf{q}^i}{\partial \mathbf{p}^j} \tag{22}$$

$$\frac{\partial {e_n}^i}{\partial \mathbf{r}^j} = \frac{\partial {e_n}^i}{\partial \mathbf{A}^i} \frac{\partial \mathbf{A}^i}{\partial \mathbf{r}^j} \tag{23}$$

$$\frac{\partial \mathbf{q}^i}{\partial \mathbf{p}^j} = \left(\frac{\partial \mathbf{q}^i}{\partial {p_x}^j} \quad \frac{\partial \mathbf{q}^i}{\partial {p_y}^j} \quad \frac{\partial \mathbf{q}^i}{\partial {p_z}^j} \right) \tag{24}$$

$$\frac{\partial \mathbf{A}^i}{\partial \mathbf{r}^j} = \begin{pmatrix} \frac{\partial \mathbf{A}^i}{\partial {r_x}^j} & \frac{\partial \mathbf{A}^i}{\partial {r_y}^j} & \frac{\partial \mathbf{A}^i}{\partial {r_z}^j} \end{pmatrix} \tag{25}$$

where **q** and **A** are expressed as Eqs. (4) and (5). To fix y-coordinate of the 3D lines labeled "on the road", the second column in (24) and (25) is computed as follows:

$$\frac{\partial \mathbf{q}^i}{\partial {p_y}^j} = \begin{cases} \mathbf{0} & (\textit{on the road}) \\ \mathbf{R}_{\mathrm{c_k v}} {\mathbf{R}_{\mathrm{vw}}}^t \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} & (\textit{on the building}) \end{cases} \tag{26}$$

$$\frac{\partial \mathbf{D}^i}{\partial {r_y}^j} = \begin{cases} \mathbf{0} & (\textit{on the road}) \\ \mathbf{R}_{\mathrm{c_k v}} {\mathbf{R}_{\mathrm{vw}}}^t \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} & (\textit{on the building}) \end{cases} \tag{27}$$

The difference between the results with or without the plane assumption is shown in Fig. 6-(c) and (d).

In the case of combining points and line segment, the cost function of points is added to Eq. (21). The cost function of the points also evaluates the re-projection error. The cost balance between points and lines are equal. However, the number of the correspondences are different. In the experiments shown in this paper, the number of inter-camera corresponding lines are 4288 (dataset 1) and 3018 (dataset 2), while the number of inter-camera corresponding points are 6945 (dataset 1) and 4062 (dataset 2), respectively. In the same way, the number of correspondences of points are generally larger, therefore the points affect the optimization than the lines.

7. Experiments

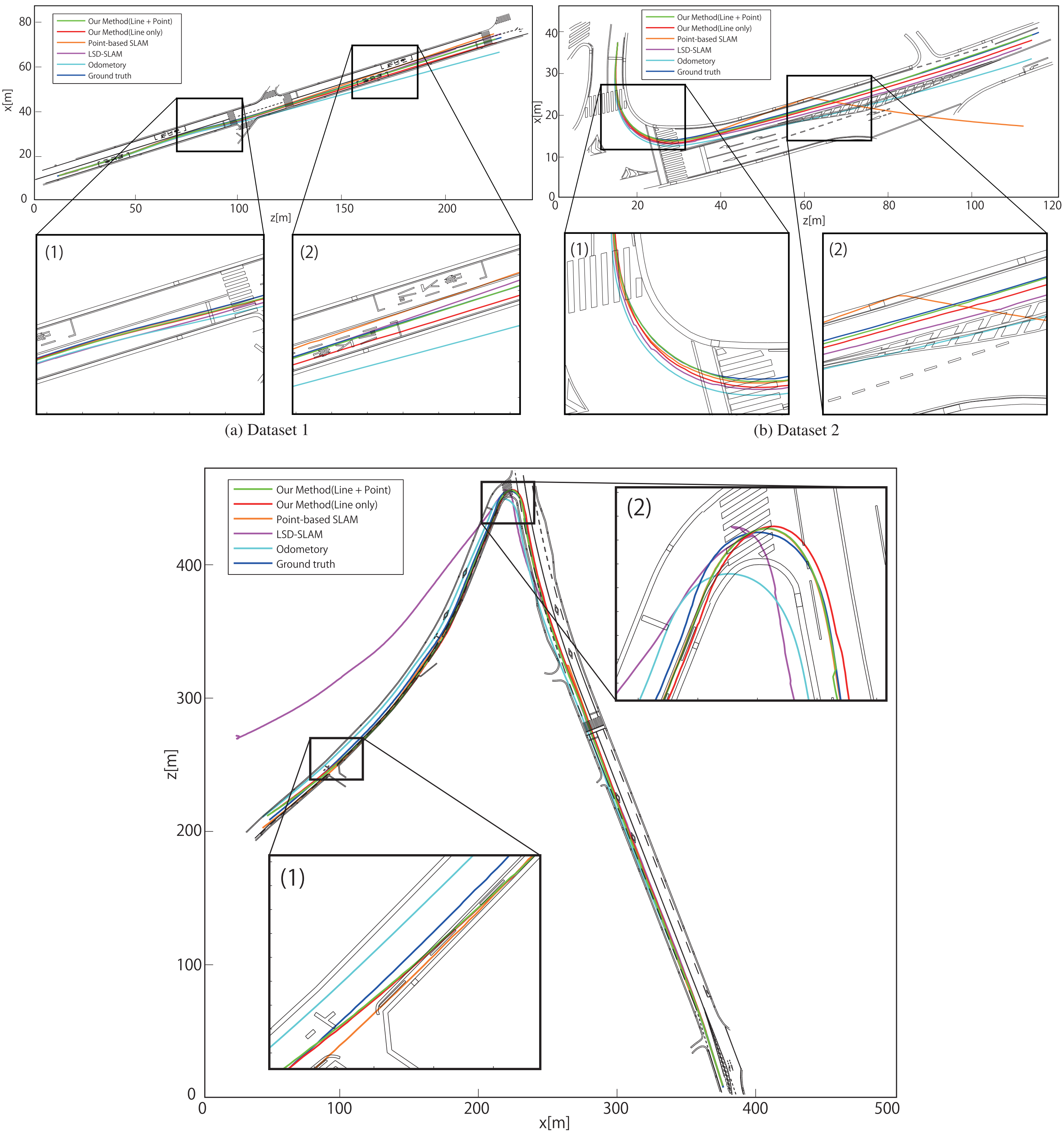
The vehicle used for experiments was equipped with various visual and motion sensors, including a high-precision RTK-GPS device (Novatel OEM615) that provides accurate motion information (to be used as a ground truth) with positioning error of less than 2 cm. We installed two cameras (Lumenera Lm225) on the top of the vehicle. An original image size is 2048 × 1088, but we resize it to 1024 × 544 image to the reduce computational complexity. The vehicle run at about 40 km/h, and the cameras captured the scene at a rate of 10 fps. Three travel datasets were collected while the vehicle was driven in an urban environment (Saiwai district, Kawasaki-City, Japan). One has 209 frames and is the scene where the vehicle goes straight, another has 202 frames and is the scene where the vehicle turns left at the intersection, and the other has 700 frames and is the scene where the vehicle travels about 1 km.

7.1 Evaluation of Localization

We evaluate the localization accuracy on the three datasets. Six methods were used for comparison: our method using points and lines, our method using lines only, point-based SLAM (previous method [3]), LSD-SLAM [23], the wheel odometry, and RTK-GPS as ground truth. LSD-SLAM is a state-of-the-art SLAM algorithm. This comparison is actually not fair because LSD-SLAM takes an image sequence with a monocular camera while our method uses two cameras, but we show this comparison with LSD-SLAM as a reference of a performance of a state-of-the-art SLAM algorithm with a monocular camera. In LSD-SLAM, the metric scale is given by the wheel odometry because this algorithm can not determine the scale.

The trajectories estimated by the six methods are shown in Fig. 7. This result indicates that our methods using both points and lines are the closest to ground truth. Our method using only lines reduces the error compared to that of the odometry. On the other hand, point-based SLAM has a large error at a certain point in zoom area (2) in Fig. 7-(b). To provide a quantitative analysis, the positional and angular errors between ground truth and the results given by each method were computed as shown in Fig. 8-(a, b, e, f, i, j). The poses of these methods for the first frame are aligned. In dataset 1 and 2, the position error of our method using points and lines is always stable within a small value (0.5 m). In dataset 3, our methods using both points and lines are the closest to ground truth with less positioning error than 1.5 meters. Although the error of the odometry are generally accumulated, the errors of our methods are hardly accumulated because the optimization is performed every time when the newest frame is obtained, so that the error of the odometry in one frame can almost be corrected. The angular errors of point-based SLAM increase at around frame 100 in Fig. 8-(b), from frame 100 in Fig. 8-(f), and at around frame 240 in Fig. 8-(j). The zoom area (2) in Fig. 7-(b) shows the trajectories around frame 100. The cause of the errors is an insufficient number of corresponding points. Only about 100 points around these frames can be obtained because the scenes are open and do not contain many building, whereas other scenes have 200 points on average. Line features can be stably detected from white lines, road markings, and curbs. This characteristic of line feature makes the two our methods more accurate than other methods. By additionally using points and lines, the accuracy of SLAM can be improved, as shown by the curve scene in Fig. 8-(f), frames 50-100 and Fig. 8-(j), frames 330-400.

To evaluate the improvement of our line-based SLAM algorithm, four cases were compared as shown in Fig. 8-(c, d, g, h, k, l): our line-based SLAM using inter- and intra-camera correspondences, our line-based SLAM using only intra-camera correspondences, standard line-based SLAM using inter- and intra-camera correspondences, and the odometry. Standard line-based SLAM estimates initial 3D lines by using line intersection of planes and does not impose the planar constraints of 3D lines on bundle adjustment. The angular errors of standard line-based SLAM is very unstable because the initial estimation of mapping of 3D line is incorrect. Comparison of standard SLAM and our methods (red plots) reveals that our initialization and optimization methods are more effective than standard line-based SLAM in the urban scene. The fact that our method



(a) Dataset 1

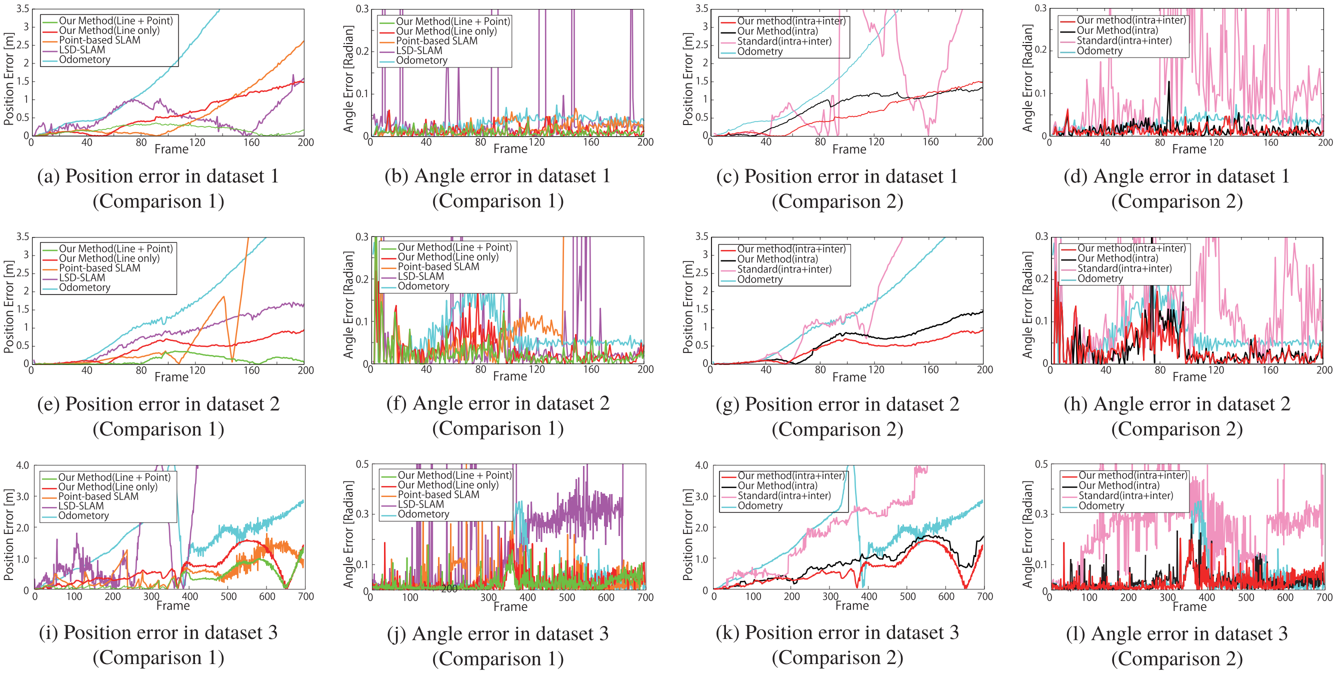
(b) Dataset 2

(c) Dataset 3

Fig. 7 Trajectory results obtained from three datasets. Comparison of result estimated by proposed method (lines plus points and only lines) with result estimated by conventional method (only points), other state-of-the-art algorithm (LSD-SLAM [23]), odometry and RTK-GPS (ground truth). The trajectories from our method (points and lines) are closest to ground truth.

(intra + inter) products slightly lower positional and angular errors than our method (intra only) confirms the effect of inter-camera correspondences on improving the accuracy.

Figure 9 is the plots of the number of detected inter-camera corresponding lines and the ratio of false matches for each frame in dataset 1 and 2. The x-axis is rear frame numbers. The total number of inter-camera correspondences in dataset 1 and 2 are 4,288 and 3,018, and the number of inliers are 4,190 and 2,898, respectively. Inter-camera correspondences are much less than the total 55,401 and 37,809



(a) Position error in dataset 1 (Comparison 1)

(b) Angle error in dataset 1 (Comparison 1)

(c) Position error in dataset 1 (Comparison 2)

(d) Angle error in dataset 1 (Comparison 2)

(e) Position error in dataset 2 (Comparison 1)

(f) Angle error in dataset 2 (Comparison 1)

(g) Position error in dataset 2 (Comparison 2)

(h) Angle error in dataset 2 (Comparison 2)

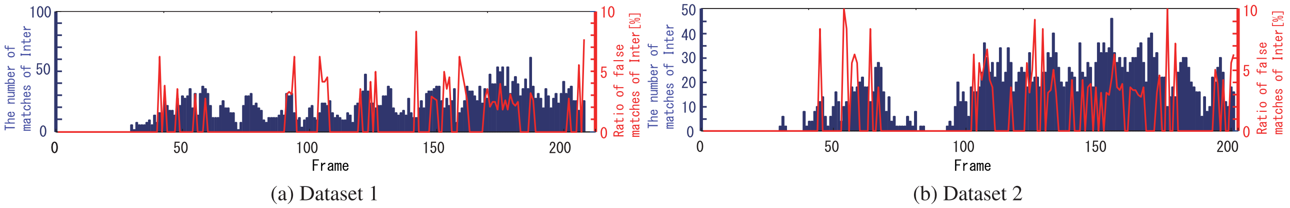
(i) Position error in dataset 3 (Comparison 1)

(j) Angle error in dataset 3 (Comparison 1)

(k) Position error in dataset 3 (Comparison 2)

(l) Angle error in dataset 3 (Comparison 2)

Fig. 8 Comparison of position error and angular error. The rows indicates the difference of dataset. “Comparison 1” focus on the comparison of the combination of lines and points. “Comparison 2” focus on the effectivity of each contribution of our method (using only line-segment). The comparisons of red and black lines show the effect of inter-camera correspondences. The comparisons of red and pink lines indicate the effect of our initialization and optimization method of 3D lines.



(a) Dataset 1

(b) Dataset 2

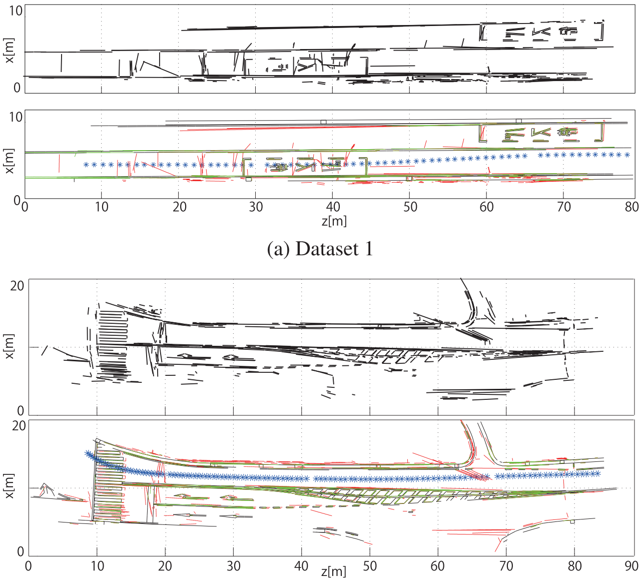
Fig. 9 The number of inter-camera correspondences and the rate of false matches in dataset 1 and 2.

intra-camera correspondence found. However, it is considered that inter-camera correspondences play a roll of connecting intra-camera correspondences of front images with ones of rear images.

7.2 Evaluation of Generated Map

Our method was used to generate a line-based road map. The endpoints of line segments on the road map can be estimated by finding intersections or nearest points between the optimized 3D line labeled “on the road” and a 3D line passing through the camera center and an endpoint of a detected line segment in the image plane. Figure 10 shows the generated line-based road map for a street with 100 meters range extracted from the dataset 1 and 2. In lower image in each dataset, the maps consisting of black line segments were made by professionals using a survey vehicle. To quantitatively evaluation of our generated maps, an inlier of line segments in our map is defined as one that have a perpendicular distance from both endpoints of the line segment to the professional map within 100 mm. In Fig. 10 green lines and red lines show inliers and outliers, respectively.

On the basis of this definition, we evaluated corresponding length and non-corresponding length in terms of true positive rate (TPR) and precision. TPR shows the rate of the length of the professional map that is matched with our generating map. In other words, if TPR is high, it can be said that many line segments of the professional map can be automatically reproduced. The precision shows the rate of the length of inliers in our generating map. If the precision is high, it can be said that few false line segments exist in our map. Table 1 shows TPR for each marking type. The second and third columns indicate TPR for the whole of dataset 1 (DS1) and dataset 2 (DS2). Each precision is 54.1% and 61.2%. These results indicate that TPR for curbs is low in DS 1, and many outliers are separated from the vehicle. One of the reasons for this separation is that the road surface is actually not plane but slightly semi-cylindrical. The shadows are also the reason that precision decreases. Most red line segments which are vertical to the travel direction in



(a) Dataset 1

(b) Dataset 2

Fig. 10 Upper images are our generated maps in (a) and (b). Lower images are comparison of our maps with ground truth. Green, red, and black lines show inliers, outliers, and ground truth, respectively.

Table 1 True positive rates for each marking type.

	Full[%]		Around car[%]	
Marking types	DS1	DS2	DS1	DS2
Lane lines	71.3	69.8	83.6	81.7
Road markings	63.0	68.0	63.0	80.8
Curbs	38.0	68.5	54.6	80.0
Total	61.5	68.0	72.8	78.0

Fig. 10-(a) indicate shadows.

The range around the vehicle was limited, and TPR and the precision were recomputed. The width of the range was set to 5 m because the width of a standard road is from 2.75 m to 3.5 m. The fourth and fifth columns of Table 1 indicate TPR. Each precision is 60.7% and 66.3%. Compared to the total TPR for the full map, that for the limited map increased by more than 10%.

For the reasons stated above, it is difficult to reproduce the complete road map using the travel dataset which was obtained in one driving. Collecting data during multiple driving of normal cars will make it possible to generate accurate maps. Future works concerning generating maps is to integrate multiple generated maps by image processing on the 2D map.

7.3 Computational Cost

We evaluate the computational cost. The computational cost of our SLAM system is expensive because it is necessary to explore all possibilities, a line segment is on a road or on a building. Experimental results shows that the matching of inter-camera correspondences takes to about 20.5 second, the matching of intra-camera correspondences takes to 38.5 second, and the optimization takes to 15.4 second. The total processing time par one frame is more than 1 minutes. This implies that our SLAM is not suitable to online application, but still useful for some applications without real-time requirement, such as generating road map.

8. Conclusion

We proposed a novel line-based SLAM using non-overlapping cameras. The proposed line segment matching algorithm can find the correspondences between different camera images even when camera viewpoints are very different from one to the other. Under the constraint that detected line segments are on buildings or on roads, the accuracy of estimating initial mapping of 3D line was improved. We also proposed a method for taking into account the constraint in bundle adjustment for accuracy improvement of SLAM. The results of localization experiments confirmed the effectiveness of inter-camera correspondences and the constraint in regard to accuracy of SLAM. Additionally, we can further improve the accuracy by using both points and line segments. Road maps were also generated the road map by our method. According to an evaluation of our generating map, TPR around the vehicle exceeding 70% is achieved. However, since it is difficult to reproduce the complete road map in one driving, it is necessary to integrate multiple maps to generate accurate maps.

References

[1] E. Teramoto, Y. Kojima, J. Meguro, and N. Suzuki, "Development of the "PRECISE" automotive integrated positioning system and high-accuracy digital map generation," R&D Review of Toyota CRDL, vol.43, pp.13–23, 2012.

[2] POSLV. Position and orientation system for land vehicles, http://www.applanix.com/media/poslvspecifications12032012.pdf

[3] A. Kawasaki, H. Saito, and K. Hara, "Motion estimation for non-overlapping cameras by improvement of feature points matching based on urban 3D structure," Proc. IEEE International Conference on Image Processing (ICIP2015), pp.1230–1234, 2015.

[4] J.M. Coughlan and A.L. Yuille, "Manhattan world: Compass direction from a single image by bayesian inference," Proc. IEEE Seventh International Conference, vol.2, pp.941–947, 1999.

[5] I. Puente, H. González-Jorge, J. Martnez-Sánchez, and P. Arias, "Review of mobile mapping and surveying technologies," Measurement, vol.46, no.7, pp.2127–2145, 2013.

[6] M. Schreiber, C. Knoppel, and U. Franke, "Laneloc: Lane marking based localization using highly accurate maps," Proc. IEEE International Conference on Intelligent Vehicles Symposium (IV2013), pp.449–454, 2013.

[7] T. Kazik, L. Kneip, J. Nikolic, M. Pollefeys, and R. Siegwart, "Real-time 6d stereo visual odometry with non-overlapping fields of view," Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR2012), pp.1529–1536, 2012.

[8] G.H. Lee, F. Faundorfer, and M. Pollefeys, "Motion estimation for self-driving cars with a generalized camera," Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR2013), pp.2746–2753, 2013.

[9] L. Kneip and H. Li, "Efficient Computation of relative pose for multi-camera systems," Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR2014), pp.446–453, 2014.

[10] R. Pless, "Using many cameras as one," Proc. IEEE Interna-

tional Conference on Computer Vision and Pattern Recognition (CVPR2003), vol.2, pp.II-587, 2003.

[11] A. Elqursh and A. Elgammal, "Line-based relative pose estimation," Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR2011), pp.3049–3056, 2011.

[12] P. Smith, I. Reid, and A.J. Davison, "Real-time monocular slam with straight lines," Proc. British Machine Vision Conference (BMVC2006), vol.6, pp.17–26, 2006.

[13] K. Hirose and H. Saito, "Fast line description for line-based slam," Proc. British Machine Vision Conference (BMVC2012), pp.83.1–83.11, 2012.

[14] T. Koletschka, L. Puig, and K. Daniilidis, "Mevo: Multi-environment stereo visual odometry using points and lines," Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2014), pp.4981–4988, 2014.

[15] Y. Lu, D. Song, and J. Yi, "High level landmark-based visual navigation using unsupervised geometric constraints in local bundle adjustment," Proc. IEEE International Conference on Robotics and Automation (ICRA2014), pp.1540–1545, 2014.

[16] H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, and W. Yu, "StructSLAM: Visual slam with building structure lines," IEEE Transactions on Vehicular Technology, vol.64, no.4, pp.1364–1375, 2015.

[17] Y. Furukawa, B. Curless, S.M. Seitz, and R. Szeliski, "Manhattan-world stereo," Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR2009), pp.1422–1429, 2009.

[18] R.G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," IEEE Transactions on Pattern Analysis & Machine Intelligence, vol.32, no.4, pp.722–732, 2010.

[19] T. Sebastian, W. Burgard, and D. Fox, "Probabilistic robotics," MIT press, 2005.

[20] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," IEEE Transactions on Pattern Analysis & Machine Intelligence, vol.29, no.6, pp.1052–1067, 2007.

[21] A. Bartoli and P. Sturm, "Structure-from-motion using lines: Representation, triangulation, and bundle adjustment," Computer Vision and Image Understanding, vol.100, no.3, pp.416–441, 2005.

[22] K. Madsen, H. Bruun, and O. Tingleff, "Methods for nonlinear least squares problems (2nd ed.)," Lecture note, pp.60, 2004.

[23] J. Engel, T. Schops, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," Proc. European Conference on Computer Vision (ECCV2014), vol.8690, pp.834–849, 2014.

Atsushi Kawasaki received B.E. and M.E. degrees in Information and Computer Science from Keio University, Japan, in 2014 and 2016, respectively. Currently, he is working at TOSHIBA corporation. His research interests are computer vision and image processing.

Kosuke Hara received B.E. and M.E. degrees from Tokyo University of Agriculture and Technology, Japan. Currently, he is working at Denso IT Laboratory corporation. Also, he is currently in Ph.D course of Graduate School of Keio University, Japan. His research interest are computer vision in vehicle technology.

Hideo Saito received his Ph.D. degree in electrical engineering from Keio University, Japan, in 1992. Since then, he has been on the Faculty of Science and Technology, Keio University. From 1997 to 1999, he joined the Virtualized Reality Project in the Robotics Institute, Carnegie Mellon University as a visiting researcher. Since 2006, he has been a full professor in the Department of Information and Computer Science, Keio University. His recent activities for academic conferences include being Program Chair of ACCV2014, a General Chair of ISMAR2015, and a Program Chair of ISMAR2016. His research interests include computer vision and pattern recognition, and their applications to augmented reality, virtual reality, and human robotics interaction.