Line-based SLAM Considering Directional Distribution of Line Features in an Urban Environment

Kei Uehara[1], Hideo Saito[1] and Kosuke Hara[2]
[1]*Graduate School of Science and Technology, Keio University, 3-14-1 Hiyoshi Kohoku-ku, Yokohama, Kanagawa, 223-0061, Japan*
[2]*Denso IT Laboratory, Cross Tower 28F, 2-15-1 Shibuya Shibuya-ku, Tokyo, 150-0002, Japan*
{kei-uehara, saito}@hvrl.ics.keio.ac.jp, khara@d-itlab.co.jp

Keywords: Line-based SLAM, Manhattan World Assumption, Road Markings, Gaussian Distribution.

Abstract: In this paper, we propose a line-based SLAM from an image sequence captured by a vehicle in consideration with the directional distribution of line features that detected in an urban environments. The proposed SLAM is based on line segments detected from objects in an urban environment, for example, road markings and buildings, that are too conspicuous to be detected. We use additional constraints regarding the line segments so that we can improve the accuracy of the SLAM. We assume that the angle of the vector of the line segments to the vehicle's direction of travel conform to four-component Gaussian mixture distribution. We define a new cost function considering the distribution and optimize the relative camera pose, position, and the 3D line segments by bundle adjustment. In addition, we make digital maps from the detected line segments. Our method increases the accuracy of localization and revises tilted lines in the digital maps. We implement our method to both the single-camera system and the multi-camera system. The accuracy of SLAM, which uses a single-camera system with our constraint, works just as well as a method that uses a multi-camera system without our constraint.

1 INTRODUCTION

Currently, advanced driver assistance systems(ADAS) has been actively researched. An autonomous car is an example of an ADAS that will enable people to go anyplace without your driving operation.To achieve that, it requires localization to calculate its' trajectory. (Teramoto et al., 2012) indicated that the accuracy of the localization required for practical use is from several dozen centimeters to several meters. For localization, one of the main methods is to use GPS. Although POSLV, one of the high-end integrated accurate positioning system, achieves accuracy within several dozen centimeters using a RTK-GPS receiver, it is not appropriate for use in the autonomous car. One reason is that the cost of RTK-GPS is very high. Another reason is that the GPS system depends on the strength of microwaves from the satellite. Thus, if the vehicle is in the tunnel, it cannot work.

Meanwhile, the visual simultaneous localization and mapping system (SLAM) has been attracting attention of researchers because it does not have these problems. To determine its position, detects feature points from the environment around the vehicle using a camera as an input. In the case that feature points can be detected fully, the accuracy of SLAM is as well as a method with laser range scanner. Since the accuracy of SLAM decreases without enough feature points, there are many studies trying to solve that problem; for example, studies that uses multi-cameras or researches detect not only points, but also lines. However, using many feature points or lines means an increasing of calculation cost, and it is fatal because the system of the autonomous car requires a real-time calculation. Increasing equipment means increasing of production costs. The less equipment attached to the vehicle, the better it is as long as the accuracy is maintained in the autonomous car.

In this paper, we propose a line-based SLAM considering a directional distribution of line features in an urban environment, which is based on the Manhattan world assumption. Based on the fact that many line segments in road markings are parallel or vertical to a vehicle's direction of travel, we define the distribution as four-component Gaussian mixture distribution. To prove our method's effectiveness, we conduct four experiments: 1) single-camera system with line-based SLAM, 2) multi-camera system with line-based SLAM, 3) single-camera system with line-and-point-based SLAM, and 4) multi-camera system with line-and-point-based SLAM. We can achieve a high accuracy of SLAM in all experiments by our method.

In addition, we simultaneously make a digital map. Although, the digital map has been used in ADAS, it must be accurate and up-to-date. Roads and road markings are destroyed frequently, so they must be updated. Our digital map can be generated while driving on streets, and it requires cameras. No other equipments are needed. SLAM often has some intrinsic or extrinsic errors, so the line segments in the map often tilt. However, it is not a problem in our method because the directional distribution is considered.

2 PREVIOUS WORKS

Our goal is to realize SLAM using the distribution of line segments. There are some SLAM works that use line and road markings. Therefore, we discuss line-based SLAM and SLAM using road markings separately in this section.

2.1 Line-based SLAM

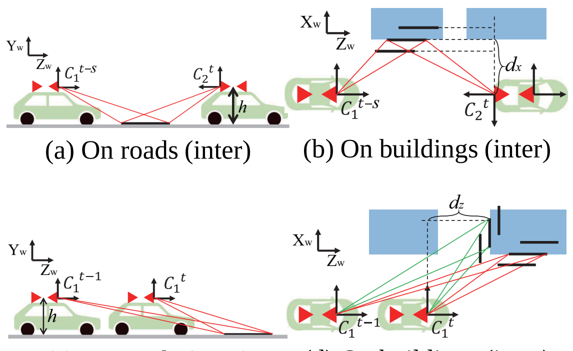
There are previous works on line-based SLAM that have taken different approaches. (Smith et al., 2006) proposed a real-time line-based SLAM. They added straight lines to a monocular extended Kalman filter (EKF) SLAM, and realized the real-time system using a new algorithm and a fast straight-lines detector that did not insist on detecting every straight line in the frame.

Another approach for the line-based SLAM is using lines and other features. (Koletschka et al., 2014) proposed a method of motion estimation using points and lines by stereo line matching. They developed a new stereo matching algorithm for lines that was able to deal with textured and textureless environments.

In addition, (Zhou et al., 2015) proposed a visual 6-DOF SLAM (using EKF) based on the structural regularity of building environments, which is called the Manhattan world assumption (Coughlan and Yuille, 1999). By introducing a constraint about buildings, the system achieved decreasing position and orientation errors.

2.2 SLAM using Road Markings

There are also previous works using road markings for SLAM for a vehicle. (Wu and Ranganathan, 2013) proposed a method for SLAM using road markings that they previously learned. They detected feature



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

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

Figure 1: Examples of intra- and inter-camera correspondences.

points from learned road markings and achieved a high accuracy in estimation of a camera pose.

(Hata and Wolf, 2014) proposed a method which detects road markings robustly for SLAM. They developed a line detector which did not affected by illumination condition by adapting Otsu thresholding method.

In addition, (Z.Tao et al., 2013) combined GPS, proprioceptive sensors, and road markings for SLAM.

3 SYSTEM OVERVIEW

We provide an overview of a line-based SLAM considering the distribution of road markings. We implement our method into both a single-camera system and a multi-camera system. We explain the method of the multi-camera system because the radical method for both the single-camera and the multi-camera systems is the almost same .

As a premise, our SLAM method requires wheel odometry. A relative camera pose and position in each frame are estimated from the data from the wheel sensors. Then, we obtain line segments from input images for each frame t by the line segment detector (LSD) algorithm. To correspond detected line segments, we consider the following three cases: matching between the front images features at frame t and $t-1$, matching between the rear image at frame t and $t-1$, and matching between the rear image at frame t and the front image at frame $t-s$. However, the viewing directions are very different, so it is hard to find matching segments, especially in the case of the third case. For a robust match, we make the feature patches around the line segments warp into other frames. Due to this warping processing, the perspective appearance of the patch resembles a target image. In addition, searching for the appropriate s value for the best match enables us to find the correspondences between a rear image and multiple front images more robustly. As long as corresponding line segments are detected, each line segment can be tracked over. Figure 1 indicates examples of matching.

Using these corresponding line segments, 3D line segments can be initially estimated using a method based on the Manhattan world assumption. After that, bundle adjustment is applied to the relative camera pose, positions and 3D line segments to optimize them. In a cost function of bundle adjustment, which is used for the optimization, we implement a new element considering the distribution of road markings, which is defined by 4-component Gaussian mixture distribution. Due to this element, the accuracy of localization improves and tilted line segments, which include some errors, in the generated map are revised.

The difference between a single-camera system and a multi-camera system is that the former does not consider the third matching case: matching between the rear image at frame t and the front image at frame $t-s$.

4 NOTATION

We explain the notations briefly before discussing our method in detail. First, we define the four coordinates W, ${C_1}^t$, ${C_2}^t$ and V^t, which indicate the world, front camera, rear camera, and vehicle coordinate systems, respectively. The relative transformations from the world coordinate system to the vehicle coordinates system at frame t are expressed as ${\mathbf{R}_{\mathrm{vw}}}^t$, ${\mathbf{T}_{\mathrm{vw}}}^t$, and these from the vehicle coordinates system to the front or the rear camera coordinate system are expressed as $\mathbf{R}_{\mathrm{c_1v}}$, $\mathbf{T}_{\mathrm{c_1v}}$, $\mathbf{R}_{\mathrm{c_2v}}$, $\mathbf{T}_{\mathrm{c_1v}}$. The relative pose and position of the front and rear cameras are calibrated beforehand. Using these notations, the projections from a world point $\mathbf{p} = (x,y,z)^{\mathrm{T}}$ to camera point $\mathbf{q} = (q_x, q_y, q_z)^{\mathrm{T}}$ are calculated as follows:

$$\mathbf{q}_k = \begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} = \mathbf{R}_{\mathrm{c_kv}}({\mathbf{R}_{\mathrm{vw}}}^t\mathbf{p} + {\mathbf{T}_{\mathrm{vw}}}^t) + \mathbf{T}_{\mathrm{c_kv}} \quad (1)$$

where k denotes camera 1 or camera 2. For this case, $k = 1$ indicates the front camera. Then, projection from a camera point $\mathbf{q}$ to a image point $\mathbf{u} = (u,v)^{\mathrm{T}}$ is calculated as follows:

$$\mathbf{u}_k = \begin{pmatrix} u \\ v \end{pmatrix} = \pi \begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} = \begin{pmatrix} q_x/q_z \\ q_y/q_z \end{pmatrix} \quad (2)$$

Secondly, the 3D line $\mathbf{L}_k$ is expressed as six-vector $({\mathbf{p}_k}^{\mathrm{T}}, {\mathbf{r}_k}^{\mathrm{T}})^{\mathrm{T}}$. The three-vectors $\mathbf{p}_k$ is a center point of the 3D line and $\mathbf{r}_k$ is the direction of the 3D line. The 2D line $\mathbf{l}_k$, which is a projection line of $\mathbf{L}_k$ in the image plane, is given as four-vector $({\mathbf{u}_k}^{\mathrm{T}}, {\mathbf{d}_k}^{\mathrm{T}})^{\mathrm{T}}$. Point $\mathbf{u}$ is calculated in the same way as the world point $\mathbf{p}$ and the vector $\mathbf{d}$ in the image plane is calculated as follows:

$$\mathbf{D}_k = \begin{pmatrix} D_x \\ D_y \\ D_z \end{pmatrix} = \mathbf{R}_{\mathrm{c_kv}}{\mathbf{R}_{\mathrm{vw}}}^t\mathbf{r}_k \quad (3)$$

$$\mathbf{d}_k = \begin{pmatrix} d_x \\ d_y \end{pmatrix} = \begin{pmatrix} \mathrm{q}_z\mathrm{D}_x - \mathrm{q}_x\mathrm{D}_z \\ \mathrm{q}_z\mathrm{D}_y - \mathrm{q}_y\mathrm{D}_z \end{pmatrix} \quad (4)$$

where $\mathbf{D}_k$ is a point of the camera coordinate system. Finally, the wheel odometry $\mathbf{x}$ is expressed as follows:

$$\mathbf{x}^{t+1} = \begin{pmatrix} x^{t+1} \\ z^{t+1} \\ \theta^{t+1} \end{pmatrix} = \mathbf{x}^t + \Delta\mathbf{x}^t + {\varepsilon_{\mathbf{x}}}^t \quad (5)$$

where $\Delta\mathbf{x}^t$ denotes its relative movement from a previous time and ${\varepsilon_{\mathbf{x}}}^t$ denotes the noise of $\Delta\mathbf{x}^t$. In the proposed method, we estimate the 3-DOF motion of the vehicle coordinates (x, z, θ) in a 2D-environment. When $\mathbf{x}^0$, which is the initial position of the vehicle, is in the same position as the origin of the world coordinate system, $\mathbf{x}^t$ equals ${\mathbf{R}_{\mathrm{vw}}}^t$ and ${\mathbf{T}_{\mathrm{vw}}}^t$. We suppose that ${\varepsilon_{\mathbf{x}}}^t$ is zero-mean Gaussian white noise with covariance $\Sigma_{\mathbf{x}}$.

$${\varepsilon_{\mathbf{x}}}^t \sim \mathcal{N}(0, \Sigma_{\mathbf{x}}) \quad (6)$$

$$\Sigma_{\mathbf{x}} = \begin{pmatrix} {\sigma_x}^2 & 0 & 0 \\ 0 & {\sigma_z}^2 & 0 \\ 0 & 0 & {\sigma_\theta}^2 \end{pmatrix} \Delta t \quad (7)$$

where σ_x, σ_z, and σ_θ denote error variances. The convariance increases in proportion to time, supposing that ${\varepsilon_{\mathbf{x}}}^t$ simply increases.

5 PROPOSED METHOD

The purpose of our method is to enhance the accuracy of line-based SLAM by bundle adjustment, which considers the distribution of the line segments. Our system consists of two parts: line matching and bundle adjustment. In this section, we explain how to realize them individually.

5.1 Line Matching

Our method has two matching algorithms: matching inter-camera correspondences and matching intra-camera correspondences, the basis of which is warping patches based on the Manhattan world assumption. We explain them individually.

5.1.1 Inter-Camera Correspondences

This method of matching is used between the front and the rear images, and is not featured with a single-camera system. The blue-colored rectangle shown in Figure 2(a) is a patch, which is 20 pixels wide and has a detected red-colored line segment in the center. The patch is transformed into a target image plane by a warp function, which considers the place where the line segments are detected. Conceivable places where they can be detected include front walls of buildings or road surfaces, as shown in Figure 1(a) and (b). Therefore, two types of warping processing are performed on all line segments in this matching algorithm.

To explain the case of the front walls of building, according to the Manhattan world assumption, a y-coordinate value of all the points in the patch in ${C_2}^t$ is h, which indicates the height of the attached camera. Therefore, the 3D point in the patch is calculated as follows:

$$\mathbf{p}_r = \begin{pmatrix} h(u_{m,2}+\alpha)/(v_{m,2}+\beta) \\ h \\ h/(v_{m,2}+\beta) \end{pmatrix} \tag{8}$$

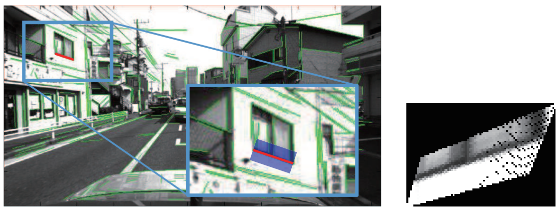
We define the coordinate of the patch as $\mathbf{u}_{m,2} = (u_{m,2}+\alpha, v_{m,2}+\beta)^{\mathrm{T}}$, where $\mathbf{u}_{m,2}$ denotes the center point of the line segment and α and β are the Euclidean distances from the center point.

In the case of the line segments detected on the road surface, the 3D point in the patch is calculated as follows:

$$\mathbf{p}_{bf} = \begin{pmatrix} d_x \\ d_x(v_{m,2}+\beta)/(u_{m,2}+\alpha) \\ d_x/(u_{m,2}+\alpha) \end{pmatrix} \tag{9}$$

The x-coordinate value of all the points in the patch in ${C_2}^t$ is d_x, which is shown in Figure 1(b), under the assumption.

The result of the warping is shown in Figure 2(b). It shows that the perspective appearance becomes similar to the target image (Figure 2(c)). Subsequently, we judge whether the warped patch and the line segments in the target image correspond or not with using an error ellipse based on the EKF proposed by (Davison et al., 2007). We use a raster scan of the error ellipse of the warped patch and calculate a zero-mean normalized cross-correlation (ZNCC) score. At the position where the ZNCC is the highest, we calculate two more values: the angle between the warped line segment and the line segments in the target image, and the distances from the endpoints of warped line segments to the line segment in the target image. If these two values are lower than the threshold, they are regarded as the correspondence.


(a) Rear image and patch (b)Warped patch

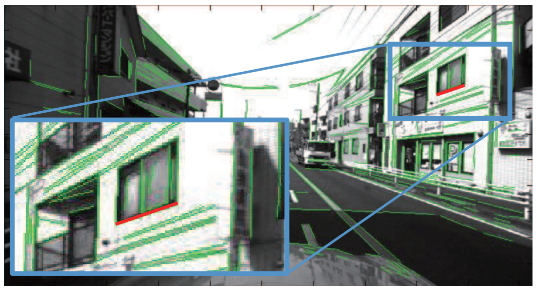

(c) Front image and a detected line segments

Figure 2: (a)A rear image with a patch of the red detected line. (b) The warped patch converted from (a). (c) The front image.

Although we propose this matching method, it has two ambiguous points: one is that we cannot precisely distinguish whether the line segments exist on buildings or the road surface, and the precise value of d_x cannot be calculated. To exclude these ambiguous elements, we test two calculations, for the building and road surface, and change d_x at regular intervals in each experiment. Then we decide the correspondence based on the highest ZNCC score.

5.1.2 Intra-Camera Correspondences

This method of matching is used between pairs of front images or pairs of rear images. There are three conceivable places where the line segments can be detected in this method: two of three are the same as the inter-camera correspondences and the last one is the side wall of buildings, as shown in Figure 1(c) and (d). The 3D point in the patch is calculated as follows:

$$\mathbf{p}_{bs} = \begin{pmatrix} d_z(u_{m,1}+\alpha) \\ d_z(v_{m,1}+\beta) \\ d_z \end{pmatrix} \tag{10}$$

Under the Manhattan world assumption, z-coordinate of all the 3D points in the patch is d_z, which is shown in Figure 1(b), so Equation 10 can be defined. As well as the inter-camera correspondences, we change d_z at regular intervals and find the best one for matching.

5.1.3 The Matching Result

The results of matching are shown in Figures 3.

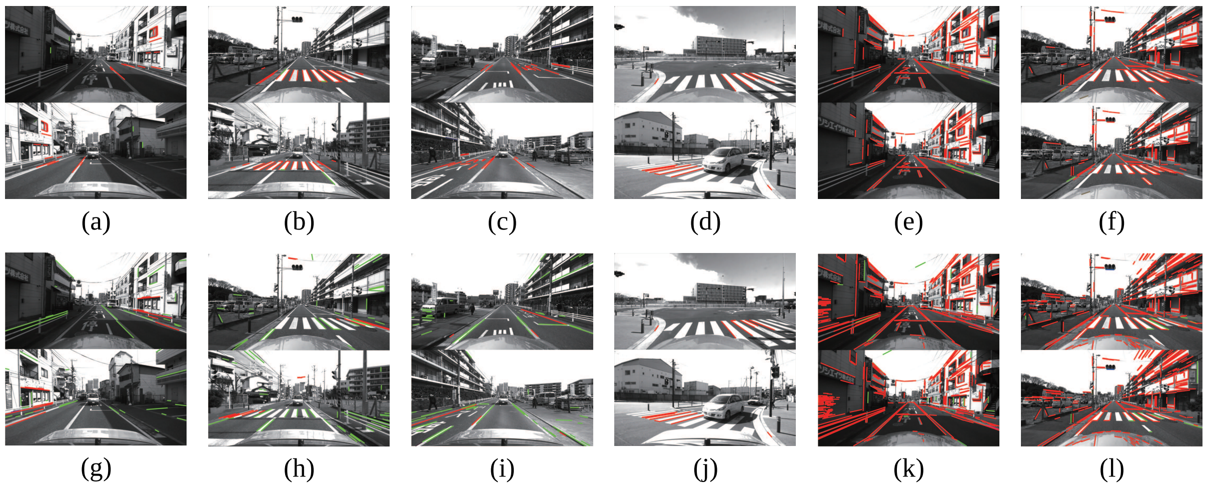


Figure 3: Examples of matching using our method (a)-(f) and LEHF (Hirose and Saito, 2012) (g)-(l). (a)-(d) and (g)-(j) are matched pairs of front and rear images. (e) (f), (k), and (l) are matched pairs of front images. The red lines and green lines show correct and incorrect matching, respectively.

Figures 3(a) to (d) indicate intra-matching correspondences with using our method, and Figure 3(g) to (j) indicate correspondences using LEHF (Hirose and Saito, 2012). Although, the enough number of the line segments cannot detected, the accuracy of the matching obviously increases.

Figures 3(e) and (f) indicate inter-matching correspondences with using our method, and Figures 3(k) and (l) indicate correspondences using LEHF (Hirose and Saito, 2012). The accuracy of them is the almost same.

Using our method, we can classify all matched line segments into *on the building walls* or *on the road*, which LEHF cannot do. This detected place data are important for the next step.

5.2 Initial Estimation of 3D Line

We use two methods to estimate the 3D line segments, which will be explained individually in this section.

First, we estimate line segments using a method based on the line of intersection of planes that passes the camera center and the line segment. Figure 4(a) shows examples of road maps generated by the 3D information calculated by this method, although most of them lie wrong place. The reason for this is because when the line segments run parallel to the travel direction, the angle between the plane passing the segments becomes too low. The 3D information create error because of that.

Secondly, we estimate the line segments using Equations (8)(9),(10) as explained in Section 5.1. In this method, we can obtain some 3D information per one line segment, because it requires only one line segment to calculate the 3D information and each line segment has some corresponded line segments. Therefore, we choose one line segment by minimum median method. An element used in the minimum median method is the re-projection error, which is calculated from the perpendicular distance from a reprojected 3D line to the endpoints of a detected line in the image plane. This method is widely used and is defined in a paper written by (Bartoli and Sturm, 2005). Figure 4(b) shows a result of this method, which is obviously better than first method.

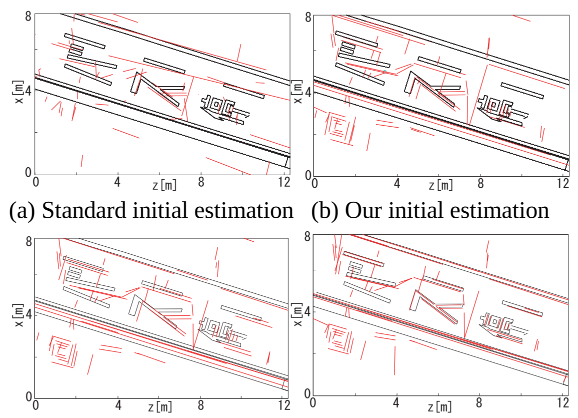
5.3 Optimization by Bundle Adjustment

We optimize the relative camera pose, position ($\mathbf{R}_{\mathrm{vw}}{}^{t}$, $\mathbf{T}_{\mathrm{vw}}{}^{t}$) and the 3D line segments $\mathbf{L}^{j}$ by bundle adjustment. We define the set of corresponding line segments Ω as :

$$\Omega = \{\omega_i = (t,k,j,p) \mid t \in \{1,\ldots,T\}, k \in \{1,2\}, j \in \{1,\ldots,J\}\ p \in \{1,2,3\}\} \tag{11}$$

where the ith line segment indicates that a 3D line is observed in a place p by camera k at frame t. In our method, the objectives are to minimize the reprojection errors of all the line segments and to minimize the angle errors of the line segments observed in the road surface. Then, the cost function is defined as following.

$$E = \mathbf{e}_l^2 + \mathbf{e}_\theta^2 = \frac{1}{2}\sum_i \sum_{n=1}^{2} d^2{}_{\perp}(\mathbf{g}_n{}^{i}, \mathbf{l}^{i}) + \frac{1}{2\sigma^2}\sum_m (e_\theta{}^{m})^2 \tag{12}$$



(a) Standard initial estimation (b) Our initial estimation

(c) Without the constraint (d) With the constraint

Figure 4: Examples of road maps.

where $d_{\perp}(\mathbf{a},\mathbf{b})$ denotes the perpendicular distance from a point $\mathbf{a}$ to a line $\mathbf{b}$ in images, ${\mathbf{g}_1}^i$ and ${\mathbf{g}_2}^i$ denote the endpoints of observed line segments, and e_θ denotes an angle error, which can be calculated using the difference between the travel direction of the vehicle and the vector of the line segments. An objective is to find the best values of ${\mathbf{R}_{\mathrm{vw}}}^t$, ${\mathbf{T}_{\mathrm{vw}}}^t$, and $\mathbf{L}^j$ to minimize E. We use the iterative non-linear Levenberg-Marquardt optimization algorithm with numerical differentiation based on a method proposed by (Madsen et al., 1999). We explain the calculation method for the reprojection error and angle error individually.

5.3.1 Reprojection Error

We use the following equation to calculate $d_{\perp}({\mathbf{g}_n}^i,\mathbf{l}^i)$:

$$e_l^i = d_{\perp}({\mathbf{g}_n}^i,\mathbf{l}^i) = \frac{d_y({g_x}^i - u^i) - d_x({g_y}^i - v^i)}{\sqrt{{d_x}^2 + {d_y}^2}} \tag{13}$$

To solve the bundle adjustment, we make Jacobian matrices made from the result of differentiated Equation (13) in the relative camera pose, position(${\mathbf{R}_{\mathrm{vw}}}^t$, ${\mathbf{T}_{\mathrm{vw}}}^t$) and the 3D line segments $\mathbf{L}^j$. The differentiation equations are expressed as follows by conforming to the chain rule:

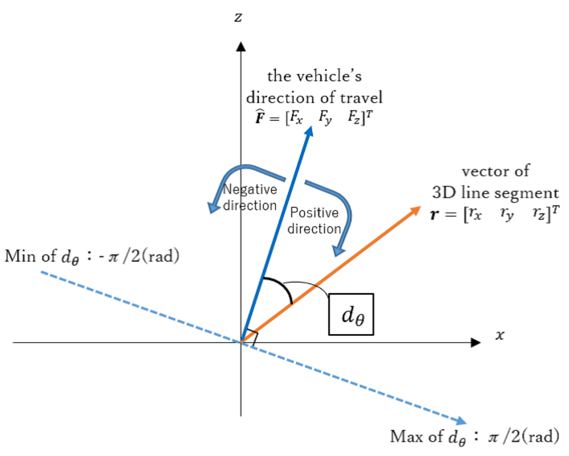
$$\frac{\partial {e_l}^i}{\partial {\mathbf{R}_{\mathrm{vw}}}^t} = \frac{\partial {e_l}^i}{\partial \mathbf{q}^i}\frac{\partial \mathbf{q}^i}{\partial {\mathbf{R}_{\mathrm{vw}}}^t} \tag{14}$$

$$\frac{\partial {e_l}^i}{\partial {\mathbf{T}_{\mathrm{vw}}}^t} = \frac{\partial {e_l}^i}{\partial \mathbf{q}^i}\frac{\partial \mathbf{q}^i}{\partial {\mathbf{T}_{\mathrm{vw}}}^t} \tag{15}$$

$$\frac{\partial {e_l}^i}{\partial \mathbf{p}^j} = \frac{\partial {e_l}^i}{\partial \mathbf{q}^i}\frac{\partial \mathbf{q}^i}{\partial \mathbf{p}^j} \tag{16}$$

$$\frac{\partial {e_l}^i}{\partial \mathbf{r}^j} = \frac{\partial {e_l}^i}{\partial \mathbf{D}^i}\frac{\partial \mathbf{D}^i}{\partial \mathbf{r}^j} \tag{17}$$

where $\mathbf{q}$ and $\mathbf{D}$ are defined as Equations (1) and (4), respectively.



z
the vehicle's direction of travel
$\hat{F} = [F_x \ F_y \ F_z]^T$
vector of 3D line segment
$r = [r_x \ r_y \ r_z]^T$
Negative direction
Positive direction
Min of d_θ : -π/2(rad)
d_θ
x
Max of d_θ : π/2(rad)

Figure 5: A schematic diagram of d_θ.

In addition, we include a geometric constraint into the cost function to enhance the accuracy of the optimization. Under the Manhattan world assumption, y-coordinate of the line segments detected on the road surface is absolutely h, so we keep it constant during the process of bundle adjustment. Since d_x and d_z have ambiguousness, we do not incorporate that constraint. Figure 4(c) and (d) are the result of mapping after bundle adjustment without or with the constraint, respectively. The figures show that the constraint works well.

5.3.2 Angle Error

First, we show that d_θ in Figure 5 is an angle between the travel direction and the vector of the 3D line segments, and it has either a positive or a negative value; the maximum value is $\frac{\pi}{2}$ and the minimum value is $-\frac{\pi}{2}$. It is calculated as follows:

$$d_\theta = \arctan\frac{F_z}{F_x} - \arctan\frac{r_z}{r_x} \tag{18}$$
$$((F_x \geqq 0 \ and \ F_z \geqq 0) \ or \ (F_x \leqq 0 \ and \ F_z \leqq 0))$$

$$d_\theta = \arctan\frac{F_z}{F_x} - \arctan\frac{r_z}{r_x} - \pi \tag{19}$$
$$((F_x \geqq 0 \ and \ F_z \leqq 0) \ or \ (F_x \leqq 0 \ and \ F_z \geqq 0))$$

where $\mathbf{F} = (F_x, F_y, F_z)$ denotes the vehicle's direction of travel. To establish the consistency of the sign of d_θ, we take π from d_θ in Equation (19). In our method, we add a new constraint about d_θ. As I discussed in section 1, we suppose that most of the road markings are parallel or vertical to the vehicle's direction of travel. Although some markings include diagonal lines, there are many markings that include parallel or vertical lines; for examples, car lanes, and markings at crosswalks. In the case of the parallel line, we

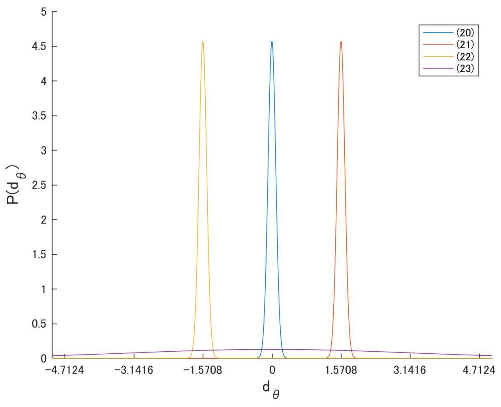


Figure 6: A four-component Gaussian mixture distribution.

assume that d_θ conforms to the Gaussian distribution. In the case of the vertical line, we assume that $d_\theta \pm \frac{\pi}{2}$ conforms to the Gaussian distribution. Figure 6 shows their distribution. Considering the diagonal lines, we define one more Gaussian distribution. It prevents the diagonal lines from being corrected to the parallel or vertical lines. The equations can be expressed as follows:

$$P_1(d^m_\theta|\mu = 0, \sigma = \sigma_\alpha) = \frac{1}{\sqrt{2\pi\sigma_\alpha{}^2}} exp(-\frac{1}{2\sigma_\alpha{}^2}(d^m_\theta)^2) \tag{20}$$

$$P_2(d^m_\theta|\mu = \frac{\pi}{2}, \sigma = \sigma_\alpha) = \frac{1}{\sqrt{2\pi\sigma_\alpha{}^2}} exp(-\frac{1}{2\sigma_\alpha{}^2}(d^m_\theta - \frac{\pi}{2})^2) \tag{21}$$

$$P_3(d^m_\theta|\mu = -\frac{\pi}{2}, \sigma = \sigma_\alpha) = \frac{1}{\sqrt{2\pi\sigma_\alpha{}^2}} exp(-\frac{1}{2\sigma_\alpha{}^2}(d^m_\theta + \frac{\pi}{2})^2) \tag{22}$$

$$P_4(d^m_\theta|\mu = 0, \sigma = \sigma_\beta) = \frac{1}{\sqrt{2\pi\sigma_\beta{}^2}} exp(-\frac{1}{2\sigma_\beta{}^2}(d^m_\theta)^2) \tag{23}$$

The numbers in the Legend of Figure 6 correspond to these equations.

Based on these equations, the cost function is calculated as follows:

$$e^m_\theta = d_\theta - \mu \tag{24}$$

This equation indicates that if the line segment is vertical, but is a little tilted, it is corrected to the accurate vertical line; and if it is parallel, but is a little tilted, it is corrected to the accurate parallel line. The value of μ is decided by the value of $P(d_\theta)$; if $P_2(d_\theta)$ is greater than other P values, μ is $\frac{\pi}{2}$.

As well as reprojection error, we make a Jacobian matrix. The differentiated equations are expressed as follows:

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

$$\frac{\partial e_\theta{}^j}{\partial r_x} = \frac{r_z}{r_x{}^2 + r_z{}^2} \tag{26}$$

$$\frac{\partial e_\theta{}^j}{\partial r_y} = 0 \tag{27}$$

$$\frac{\partial e_\theta{}^j}{\partial r_z} = -\frac{r_x}{r_x{}^2 + r_z{}^2} \tag{28}$$

In addition, we obtain the best value of σ_α and σ_β by changing them at regular intervals for the best optimization in each experiment.

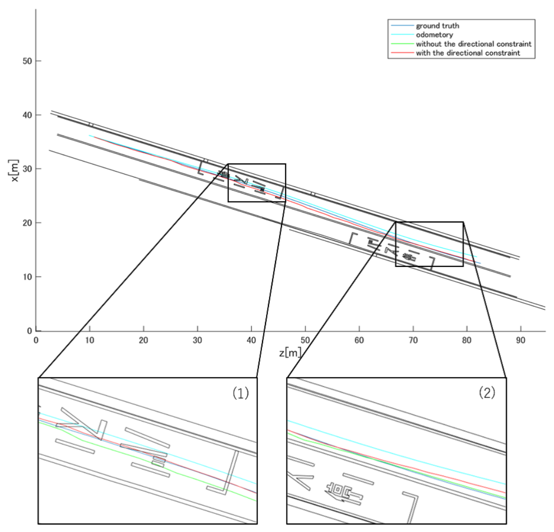
6 EXPERIMENT

In this section, we introduce a practical experiment that uses a real vehicle driving in an urban environment. Two cameras and a RTK-GPS, which can get high accuracy of self-position, are attached to the vehicle. The frame rate of the camera is 10 fps and we prepare two datasets: one is a straight scene that has 72 frames and another is curve scene that has 200 frames. We use the GPS as a ground truth. We evaluate the accuracy of localization and mapping individually.

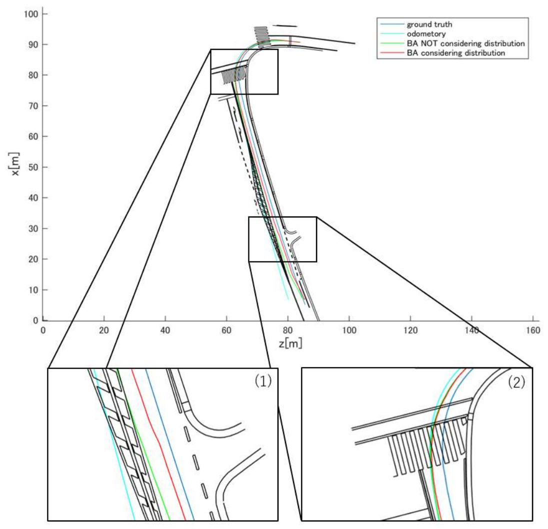
6.1 Evaluation of Localization

In our experiment, we apply our method to four cases; 1) single-camera system with line-based SLAM (called line (S)), 2) single-camera system with point-and-line based SLAM (point-line (S)), 3) multi-camera system with line-based SLAM (line (M)), and 4) multi-camera system with point-and-line based SLAM (point-line (M)). We check how much the accuracy of localization improves when the directional distribution of road markings is considered in each cases.

Figure 7 shows trajectories of the vehicles in the case of line (S) in two datasets. We compare four types of data in each case; ground truth, odometry, optimized data without the directional constraint, and optimized data with the directional constraint. A closeup area (1) of Figure 7(a) and (b) indicates that the optimized data with the constraint is the closest to the ground truth and it has higher accuracy than that without the constraint. However, (2) of Figure 7(a) indicates the accuracy decreases by adding the

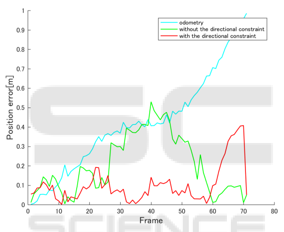


(a) Dataset 1

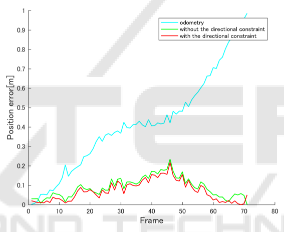


(b) Dataset 2

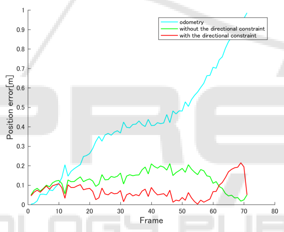
Figure 7: The trajectory results obtained from datasets 1 and 2 in the experiment of line (S). Comparison of the results estimated by ground truth, odometry, optimized data without the constraint, and optimized data with the constraint



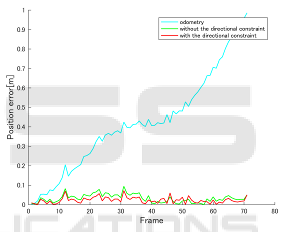
(a) line (S)



(b) line (M)

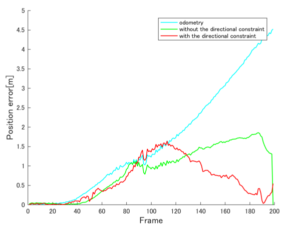


(c) line-point (S)

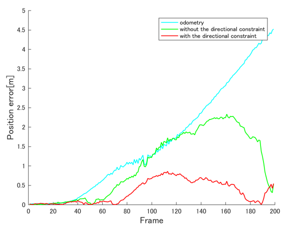


(d) line-point (M)

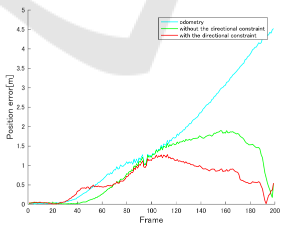
Figure 8: Comparison of position error in each frame (dataset 1).



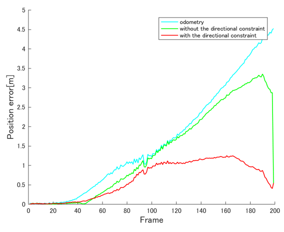
(a) line (S)



(b) line (M)



(c) line-point (S)

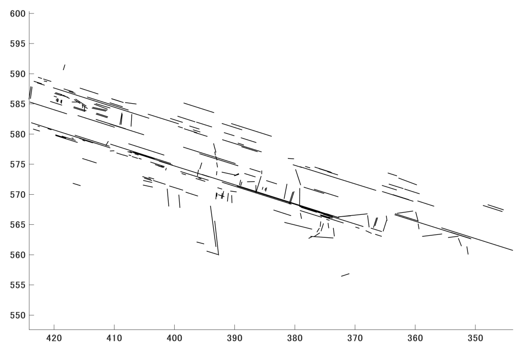


(d) line-point (M)

Figure 9: Comparison of position error in each frame (dataset 2).

Table 1: The sum of the position error and the improvement rate.

	Dataset 1			Dataset 2		
	Without the constraint [m]	With the constraint [m]	Improvement rate [%]	Without the constraint [m]	With the constraint [m]	Improvement rate [%]
line (S)	13.57	6.52	52.0	176.11	107.32	39.1
line (M)	6.22	5.54	10.9	205.56	65.16	68.3
line-point (S)	8.45	4.32	48.9	186.18	125.96	32.3
line-point (M)	2.62	1.58	39.7	267.90	129.20	51.8


(a) Dataset 1

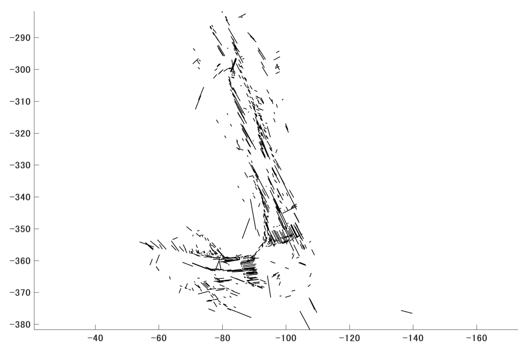
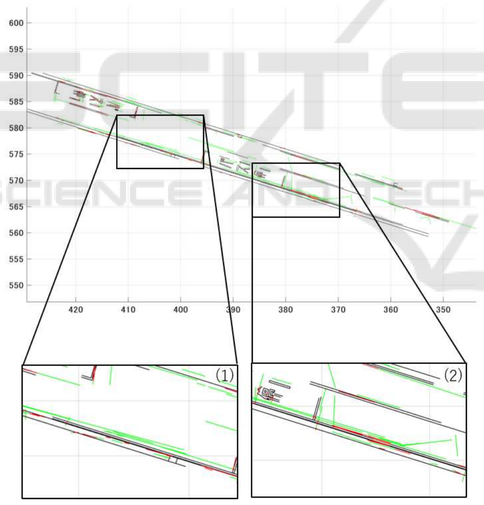

(b) Dataset 2

Figure 10: Digital maps made from detected line segments.

Table 2: The rate of inlier lines

	Dataset 1			Dataset 2		
	Without the constraint [%]	With the constraint [%]	Change [%]	Without the constraint [%]	With the constraint [%]	Change [%]
line (S)	28.5	47.5	+19	40.1	35.9	-4.2
line (M)	35.1	41.6	+6.5	39.6	36.9	-2.7
line-point (S)	39.1	49.7	+10.6	41.5	33.4	-8.1
line-point (M)	49.6	52.8	+3.2	42.2	39.1	-3.1


(a) Without the constraint

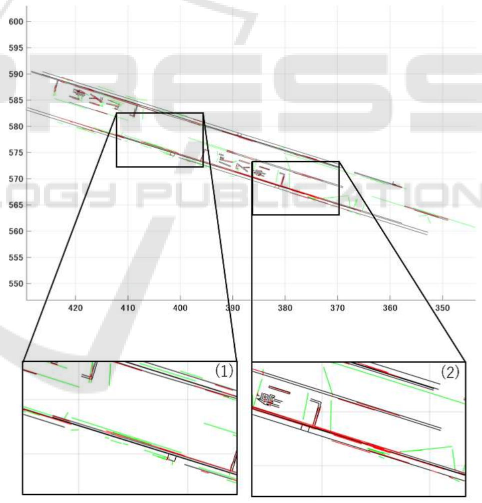

(b) With the constraint

Figure 11: Mapping results in line (S) Red lines indicate inliar lines and green lines indicate outlier lines.

constraint. The reason for that is because the vehicle cannot detect enough line segments. Actually, there are few line segments in the area.

Figure 8 and Figure 9 provide a quantitative analysis of the accuracy of localization. They are position errors, which indicate a perpendicular distance to the ground truth. In addition, Table 1 shows the sum of the position errors in each case and the rate of improvement in percentage. The accuracy improves in all experiments by considering the constraint. For dataset 1, the accuracy improves about 50% in the single-camera experiment. The accuracy of line (S) with the constraint is as well as line (M) without it. For dataset 2, the accuracy improves, especially in the multi-camera experiments. Since our SLAM method is based on the Manhattan world assumption, it is not appropriate for the curving scene because the assumption does not stand up well . However, by using the directional constraint, the error generated in the curving scene is revised. It may be a reason for the high improvement rate seen in dataset2.

6.2 Evaluation of Mapping

The detected line segments can be used to make digital maps. Figure 10 show the results of the digital map. To produce a quantitative analysis, we judge whether the generated line segments are inlier or outlier. Inlier lines are defined as the perpendicular distance from the endpoints of the generated lines to a professional line within 100 mm. Table 2 shows the rate of inliers in each experiment. The results of dataset 1 become more accuracy, while that of dataset 2 gets worse. The reason is that dataset 2 has more diagonal lines, which are collected to parallel or vertical lines although it does not. Dataset 1 has many vertical and parallel lines, so the rate of inliers increases in any experiment. Figure 11 shows whether the line segments in the digital map are whether inlier or outlier in line (S) of dataset 1 . Red lines indicate inlier lines and green lines indicate outlier lines. Tilted lines in Figure 11(a) is revised, so they change to green lines in Figure 11(b).

7 CONCLUSIONS

In this paper, we propose a line-based SLAM considering the directional distribution of line features in an urban environment. We regard the directional distribution of road markings as a combination of Gaussian distribution, and define a new constraint to a cost function of bundle adjustment. In the practical experiment, we prove that the accuracy of SLAM improves in all cases. Due to our method, the single-camera SLAM is as accurate as the multi-camera SLAM. In addition, we make digital maps from the detected line segments. Tilted lines are revised by our method, but diagonal lines are badly corrected in some cases. We will improve our method to apply to other cases.

REFERENCES

Bartoli, A. and Sturm, P. (2005). Structure-from-motion using lines:representation, triangulation, and bundle adjustment. In *Computer Vision and Image Understanding*. Elsevier.

Coughlan, J. M. and Yuille, A. L. (1999). Manhattan world: Compass direction from a single image by bayesian inference. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. IEEE.

Davison, A. J., Reid, I. D., Molton, N. S., and Stasse, O. (2007). Monoslam: Real-time single camera slam. In *Pattern Analysis and Machine Intelligence, IEEE Transactions on*. IEEE.

Hata, A. and Wolf, D. (2014). Road marking detection using lidar reflective intensity data and its application to vehicle localization. In *Intelligent Transportation Systems(ITSC), 2014 IEEE 17th International Conference on*. IEEE.

Hirose, K. and Saito, H. (2012). Fast line description for line-based slam. In *Proceedings of the British Machine Vision Conference*. BMVA.

Koletschka, T., Puig, L., and Daniilidis, K. (2014). Mevo: Multi-environment stereo visual odometry using points and lines. In *Intelligent Robots and Systems (IROS), 2014*. IEEE.

Madsen, K., Bruun, H., and Tingleff, O. (1999). Methods for non-linear least squares problems. In *Informatics and Mathematical Modelling, Technical University of Denmark*. Citeseer.

Smith, P., Reid, I. D., and Davison, A. J. (2006). Real-time monocular slam with straight lines. In *Proceedings of the British Machine Vision Conference*. BMVC.

Teramoto, E., Kojima, Y., Meguro, J., and Suzuki, N. (2012). Development of the “precise” automotive integrated positioning system and high-accuracy digital map generation. In *R&D Review of Toyota CRDL*. IEEE.

Wu, T. and Ranganathan, A. (2013). Vehicle localization using road markings. In *Intelligent Vehicles Symposium(IV), 2013 IEEE*. IEEE.

Zhou, H., Zou, D., Pei, L., Ying, R., Liu, P., and Yu, W. (2015). Structslam: Visual slam with building structure lines. In *Vehicular Technology, IEEE Transactions on*.

Z.Tao, Bonnifait, P., V.Fremont, and J.Ibanez-Guzman (2013). Mapping and localization using gps, lane markings and proprioceptive sensors. In *Intelligent Robots and Systems(IROS), 2013 IEEE/RSJ International Conference on*. IEEE.