Paper:

Vehicle Localization Based on the Detection of Line Segments from Multi-Camera Images

Kosuke Hara* **and Hideo Saito****

*Research & Development Group, Denso IT Laboratory
CROSSTOWER 28F, 2-15-1 Shibuya, Shibuya-ku, Tokyo 150-0002, Japan
E-mail: khara@d-itlab.co.jp
**Department of Information and Computer Science, Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan
E-mail: saito@hvrl.ics.keio.ac.jp
[Received June 20, 2015; accepted September 25, 2015]

For realizing autonomous vehicle driving and advanced safety systems, it is necessary to achieve accurate vehicle localization in cities. This paper proposes a method of accurately estimating vehicle position by matching a map and line segment features detected from images captured by a camera. Features such as white road lines, yellow road lines, road signs, and curb stones, which could be used as clues for vehicle localization, were expressed as line segment features on a two-dimensional road plane in an integrated manner. The detected line segments were subjected to bird's-eye view transformation to transform them to the vehicle coordinate system so that they could be used for vehicle localization regardless of the camera configuration. Moreover, an extended Kalman filter was applied after a detailed study of the line observation errors for realizing real-time estimation. Vehicle localization was tested under city driving conditions, and the vehicle position was identified with sub-meter accuracy.

Keywords: localization, multi camera system, line segment detection, autonomous driving

1. Introduction

Environment recognition using on-vehicle sensors has been studied for realizing autonomous vehicle driving and advanced safety systems. The demand for autonomous driving [1] has been increasing in the recent years; lane detection is necessary to determine the vehicle's drive trajectory for autonomous driving. For simple lane structures such as those of highways, conventional white line detection method [2] can be used for lane detection. However, lane detection in cities is difficult owing to challenges such as complicated lane structures and road signs, influence of occlusion by vehicles and people, and difficulty in recognition because of insufficient resolution of sensors in a wide crossing. In particular, barrier-free sidewalks slope toward roads, and it is difficult to identify the boundary of the roads simply by tracing the curb height. Therefore, the conventional method [1] for autonomous driving in cities is to keep the vehicle on a lane by preparing a road map with detailed information of lanes and signs and using accurately estimated vehicle position [1]. Highly accurate GPS is used for accurate vehicle localization. In fact, real-time kinematics GPS and GPS employing correction data from quasi-zenith satellites with accuracy of several centimeters have been used in autonomous driving experiments [3]. However, the use of such GPS involves problems such as high cost, unstable performance in cities with tall buildings that can affect satellite signal, and direct influence from map deviation caused by plate movement. To solve these problems, a method of estimating the relative position of a vehicle on a map by matching measurement data from on-vehicle sensors, such as cameras and laser scanners, and road maps and images has been proposed [4–15]. This technology identifies the place where the measurement data from the on-vehicle sensors match the landform on the maps or images as the vehicle position. Unlike technologies using accurate GPS or white line detection, this method takes advantages of many clues in a complicated environment, thus improving the accuracy. In this paper, we propose a method of real-time vehicle localization by matching a map and images from on-vehicle multi-cameras.

2. Related Studies

Vehicle localization methods that involve matching information from on-vehicle sensors with road maps and image databases can be classified into three types, depending on the data used. One type of localization methods employs aerial images (or satellite images), while another type employs images captured by on-vehicle cameras in the past. The third type of methods employs line maps or spline curve maps.

In vehicle localization using aerial images [4], vehicle position is identified by matching road paint in on-vehicle camera images with that in aerial images. However, the problem in this method is that a large amount

of high-quality data has to be accumulated before the method can be applied commercially. A commercial aerial photograph is created by connecting many images, and a connection error can result in the deviation of the absolute position of vehicle. Further, in aerial images, stopped/running vehicles and shadows of buildings may cause occlusion, making it difficult to maintain stable quality; hence, the definition of quality standards itself may be a problem.

The method in which past on-vehicle camera images [5, 6] are used can handle not only roads, but also buildings and other objects around roads. In general, the association between image features is robust, but this method has the same problem in terms of data volume as that in the case of the aerial photograph method. In addition, it is difficult to check the content of descriptors visually and assure the quality. In the field of robot research, a combination of laser scanner and camera has been studied [7, 8], but it is difficult to apply the results of these studies because the features of road environments are different from those applied in the studies.

In the method employing road maps containing line segments and spline curves, the maps of white road lines, yellow road lines, road signs, street gutters, curb stones, etc. are used. Creation of such complicated maps became possible because of the development in surveying technology [16] brought about by the Mobile Mapping System (MMS). Since these maps can be considered as an extension of the maps used in conventional car navigation systems, handling of data format and quality standards is rather easy. Autonomous vehicle driving and lane keeping assist are examples of the applications of vehicle localization. For these applications, road boundary information such as white road lines, curb stones, and road signs are necessary to determine the driving trajectory [17]. This kind of information is saved in the form of line segments to determine the trajectory. Since these data are often required for many applications, there are high expectations that maps will be developed with the data. Some vehicle localization methods use such road maps. For example, Lane LOC [9] uses stereo cameras to detect individual white road lines and curb stones and compares them with those in the maps to identify vehicle position. Similarly, Nedevschi et al. [10] used stereo cameras to detect individual white road line boundaries, curb stones, and stop lines. However, the problem with these methods is that the detection system configuration is complicated because different detectors had to be prepared to detect different targets such as white lines and curb stones. The use of stereo cameras has an advantage in that they can detect curb stones and road surface; however, the disadvantage in using them is that the cost increases if many stereo cameras are required. In another study [11], a single feature detector with a single eye was used. However, in this study, a white load line detection technology was used; this technology might not be able to handle cases in which a vehicle makes a turn at a complicated crossing. On the other hand, in some methods, feature matching was performed without using a features detector: these include a method of using a laser scanner's reflection intensity for features matching [12] and a method of using the structure tensor from a single-eye camera for features matching [13]. Both methods optimize the vehicle position by using a particle filter and have an advantage in that the system configuration is relatively simple than in the case in which different detectors are required to detect different features. However, particle filters often require a high computational load, which is a disadvantage for implementing filters in a vehicle. Actual autonomous vehicle driving in cities [15] often involves a combination of the method of using images taken from the vehicle [6] and the method of using line segment maps [9]. Although front and rear cameras are used, the two methods used different cameras and identified the vehicle position individually; the identified positions were merged to one by using a vehicle model. Therefore, the algorithm of this method is not suitable for a multi-camera configuration. The method closest to the present one was studied by Yu et al. [14]. As in the present method, their method used line segment features. However, since their method expresses vehicle position and orientation using six degrees of freedom, point features and vertical edge features of buildings are also required. A vehicle moves on a two-dimensional plane though the plane may have a gradient; hence, it has only three degrees of freedom. Therefore, the method can be simplified. Further, in this method, optimization is realized through RANSAC by using a complicated combination of the lines and points, and hence, the frame rate is slow.

This paper proposes a method of accurate vehicle localization based on comparison of line segment maps and line segment features extracted from multi-camera images. Different road features such as white road lines, yellow road lines, road signs, curb stones, etc., which can be used as clues for vehicle localization, are expressed as line segment features on a two-dimensional road plane in an integrated manner. The detected line segments are transformed by bird's-eye view transformation to the vehicle coordinate system in order to make the processing independent of the camera configuration. In addition, an extended Kalman filter is used after a detailed examination of line observation errors to realize real-time estimation. The accuracy of vehicle localization was evaluated through driving experiments in cities to ensure sub-meter accuracy.

3. Vehicle Localization by Multi-Camera Images

In this method, line segment features are extracted and compared with those obtained from maps to predict the vehicle position and correct the prediction (**Fig. 1**). A series of images from many cameras mounted on a vehicle and the rear wheel speed are used as the inputs. The output is the position of the vehicle on the map. Three degrees of freedom of the position and orientation are

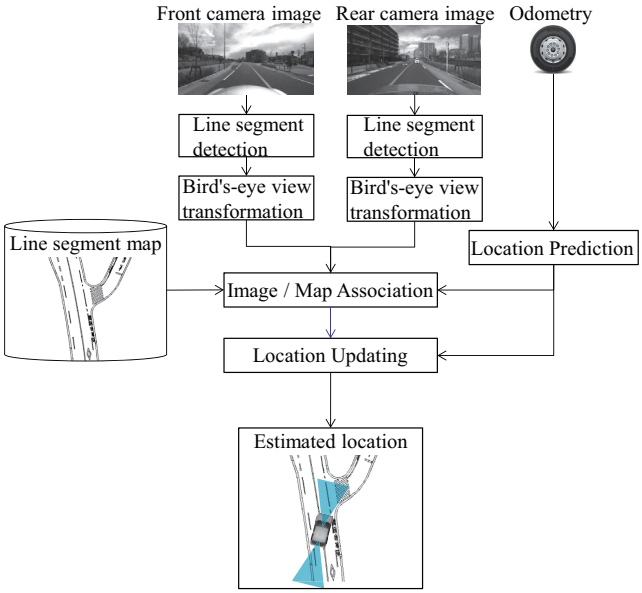


Fig. 1. The flowchart of the localization system.

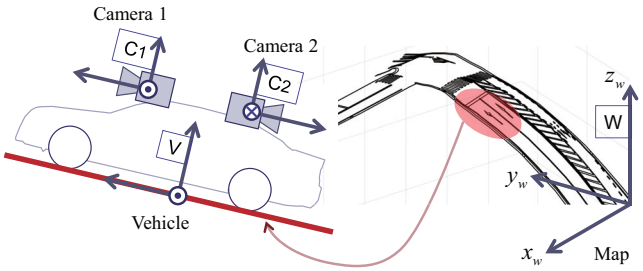


Fig. 2. Setting of the coordinate system.

estimated. First, line segments are extracted from each camera image and transformed into the vehicle coordinate system by bird's-eye view transformation. Errors are calculated by associating the transformed line segments and those in the maps, and the extended Kalman filter is used to update the vehicle position. It is assumed that the initial location of the vehicle can be obtained from GPS or by some other method; the present study focuses on updating the location of a running vehicle.

Front and rear cameras are used. Cameras with wide angle lenses have advantages in terms of camera rotation operations and are recommended for many SLAM studies [18]. Since vehicle localization is also necessary when a vehicle makes a turn at a crossing, a multi-camera configuration with no overlap in the view fields of the cameras, which can therefore be regarded as camera with a wide angle lenses, will make a large contribution to realizing highly accurate vehicle localization. Further, a previous study [19] has shown that when front and rear cameras are placed symmetrically, the accuracy is better than in the case of other camera configurations. In practical use, relatively high performance cameras are usually mounted on the front and rear sides of a vehicle as a safety system because the vehicle mostly moves forward and backward. Therefore, the present camera configuration is a practical choice. Each step of processing is explained below in detail.

3.1. Design of Coordinate Transformation

Here, the features obtained from cameras mounted on the vehicle and the design of handling three-dimensional line segment maps are explained. In the present method, a map coordinate system, a vehicle coordinate system, and camera coordinate systems are used (**Fig. 2**). First, a three-dimensional point $\mathbf{p}_{c_n}$ in the n-th camera coordinate system is transformed to a point $\mathbf{p}_v$ in the vehicle coordinate system through rotation and translation transformations that represent camera location. $\mathbf{R}_{vc_n}$ and $\mathbf{t}_{vc_n}$ are calibrated in advance and can be used as known parameters.

$$\mathbf{p}_v = \mathbf{R}_{vc_n}^T(\mathbf{p}_{c_n} - \mathbf{t}_{vc_n}) \quad \ldots\ldots\ldots\ldots (1)$$

Next, the point $\mathbf{p}_v$ in the vehicle coordinate system is transformed to a point $\mathbf{p}_w$ in the map coordinate system. In the present method, three degrees of freedom are estimated at time t by introducing the yaw rotation of the vehicle $\mathbf{R}_{vw}^t$ and the translation on a plane $\mathbf{t}_{vw}^t$. An actual road will have a gradient, and hence, the vehicle position and orientation have to be defined with six degrees of freedom. However, considering too many variables in the estimation will cause degradation of accuracy. Therefore, based on the fact that the line segment map has three-dimensional coordinates, the vehicle was assumed to move on a plane, which was an approximation of a local area in the maps. In other words, taking into account the width of a general street, a local map area of scale 3.5 m around the vehicle is assumed to be a plane, and the gradient $\mathbf{R}_s^t$ of roll and pitch is calculated. The vehicle position in the height direction can also be calculated from the plane. Then the coordinate transformation can be expressed by the following formula by taking into account the road gradient.

$$\mathbf{p}_w = \mathbf{R}_s^t \mathbf{R}_{vw}^t \mathbf{p}_v + \mathbf{t}_{vw}^t \quad \ldots\ldots\ldots\ldots (2)$$

3.2. Prediction with the Vehicle Movement Model

In this section, a model of the vehicle movement is described, and prediction of the movement state and an error model are explained. Many recent vehicles have sensors to detect right and left rear wheel speeds v_R^t and v_L^t, respectively. Application software can utilize the speed data through a CAN network. Odometry employs such data for estimating vehicle motion, i.e., for calculating the estimated vehicle position at the time when the image was captured. Let $\mathbf{x}^t = (x^t, y^t, \theta^t)$ be the vehicle position and orientation at time t with rotation $\mathbf{R}_{vw}^t$ and translation $\mathbf{t}_{vw}^t$. Then the estimated vehicle position is given by the following relations:

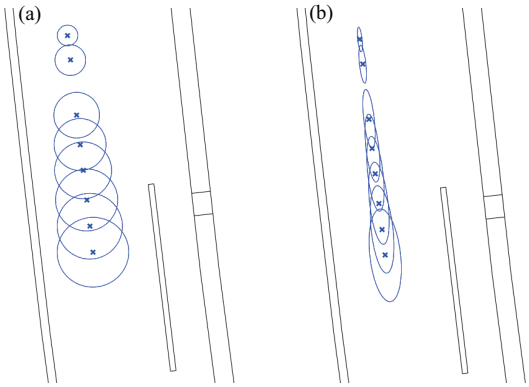


Fig. 3. The estimation error in vehicle movement.

$$\bar{\mathbf{x}}^t = \mathbf{f}(\mathbf{x}^{t-1}, \mathbf{u}^t) + \boldsymbol{\varepsilon}^t \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (3)$$

$$\mathbf{f}(\mathbf{x}^{t-1}, \mathbf{u}^t) = \mathbf{x}^{t-1} + \begin{bmatrix} \frac{v^t}{\omega^t}\cos\left(\theta^{t-1}\right) - \frac{v^t}{\omega^t}\cos\left(\theta^{t-1} + \omega^t \Delta t\right) \\ -\frac{v^t}{\omega^t}\sin\left(\theta^{t-1}\right) + \frac{v^t}{\omega^t}\sin\left(\theta^{t-1} + \omega^t \Delta t\right) \\ \omega^t \Delta t \end{bmatrix}$$

Function $\mathbf{f}$ is given by the same formula as the one used for a two-wheel robot model [20]. It is also assumed that the estimation error $\boldsymbol{\varepsilon}^t$ follows a Gaussian distribution with average $\mathbf{0}$ and covariance $\mathbf{Q}^t$. Control variable $\mathbf{u}^t = (v^t, \omega^t)^T$ includes the vehicle speed $v^t = (v_R^t + v_L^t)/2$ and the angular velocity $\omega^t = (v_R^t - v_L^t)/2d_r$, both of which are calculated from the rear wheel speed. Further, d_r is the distance between the wheels.

In general, estimation error $\mathbf{Q}^t$ is often represented by a diagonal matrix as the parameter. However, if $\mathbf{Q}^t$ is designed to be a diagonal matrix, the error ellipse expands as circle on the road plane; this behavior is very different from that observed during actual vehicle movement (**Fig. 3(a)**). Therefore, the major errors were assumed to be contained in the observed control variable and to be extended as the vehicle moved (**Fig. 3(b)**). This assumption is expressed by the following formula in the first order of Jacobian $\partial\mathbf{f}/\partial\mathbf{u}^t$.

$$\mathbf{Q}^t = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}^t}\mathbf{Q}_u\frac{\partial \mathbf{f}}{\partial \mathbf{u}^t}^T\right)\Delta d + \mathbf{Q}_c \Delta t \quad \ldots\ldots \quad (4)$$

Here, Δd is the distance moved by the vehicle in unit time, and Δt is the time difference.

$\mathbf{Q}_u$ is a diagonal matrix that represents the errors in the control variables, and $\mathbf{Q}_c$ is a diagonal matrix that represents minor deviation errors from the model.

3.3. Line Extraction from Multi-Camera Images

Line segments were extracted using LSD (Line Segment Detector) [21], which is a highly accurate high-speed detection method, from multi-camera images obtained from the on-vehicle cameras. LSD was also used previously for a study of the real-time SLAM [22]. **Fig. 4** shows the results of the application of this method to the on-vehicle camera images. The blue lines indicate the detected lines. The figure shows the line segments of the white road lines and sidewalks and also those of street gutters and curb stones. Thus, line segment detection can detect most of the characteristic objects on the road, and this method can be used for vehicle localization. Since the cameras were mounted horizontally on the vehicle, objects located at a certain height or higher were not included in the processing. In addition, the lines resulting from the reflection of the vehicle itself were identified and removed from the images.

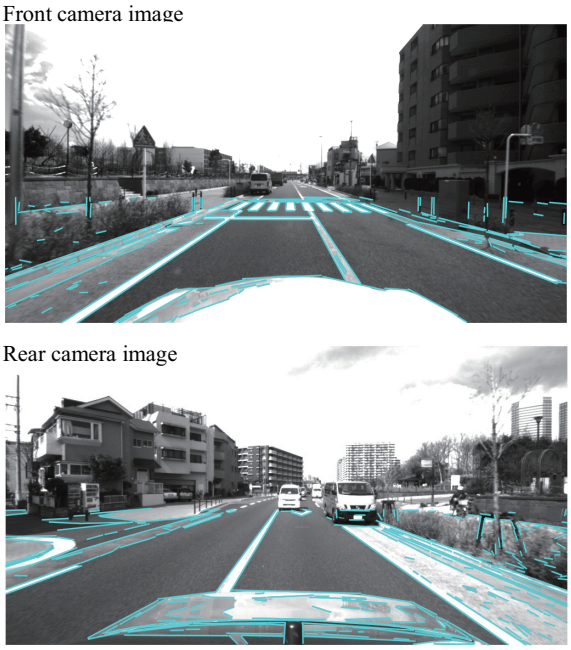


Fig. 4. Line detection with multi-camera images.

3.4. Bird's-Eye View Transformation of Lines in Images

Line segments detected from the cameras were handled in the vehicle coordinate system to ensure that changes in the camera configuration did not affect the subsequent processing. For this purpose, the line segments in the images were projected onto the vehicle coordinate system by bird's-eye view transformation, and their error distribution was examined. The rotation and translation representations $\mathbf{R}_{vc_n}$ and $\mathbf{t}_{vc_n}$ of the camera positions and the internal parameters of the cameras were assumed to be known, and the road surface was assumed to be flat in the vicinity of the vehicle. The start point $\mathbf{n}_s$ and end point $\mathbf{n}_e$ of a line segment in a normalized image were transformed by the bird's-eye view transformation $\mathbf{b}$ to $\mathbf{q}_s$ and $\mathbf{q}_e$, respectively, in the vehicle coordinate system. The start and end points are transformed in the same way, and hence, we write them as $\mathbf{q} = \mathbf{b}(\mathbf{n})$ omitting the subscripts. Then the transformation formula is given as below.

$$\mathbf{R}_{vc_n} = \{r_{ij}\}, \mathbf{t}_{vc_n} = [t_x, t_y, t_z]^T \quad \ldots\ldots\ldots \quad (5)$$

$$q_x = \frac{-n_v t_y(n_u r_{33}-r_{13})+t_z(n_u r_{33}-r_{13})+n_u t_y(n_v r_{33}-r_{23})-t_x(n_v r_{33}-r_{23})}{(n_v r_{31}-r_{21})(n_u r_{33}-r_{13})-(n_u r_{31}-r_{11})(n_v r_{33}-r_{23})}$$

$$q_y = \frac{-n_v t_y(n_u r_{31}-r_{11})+t_z(n_u r_{31}-r_{11})+n_u t_y(n_v r_{31}-r_{21})-t_x(n_v r_{31}-r_{21})}{(n_v r_{33}-r_{23})(n_u r_{31}-r_{11})-(n_u r_{33}-r_{13})(n_v r_{31}-r_{21})}$$

For realizing position correction with an extended Kalman filter, it is important to develop a model of observation errors for the detected line segments. The error model obtained after the bird's-eye view transformation of the line segments was examined in terms of the detected position and length of the line segments. Note that the camera lens is the major source of errors in position detection using images. In general, the error increases as the distance of the area from the center increases. The errors persist despite distortion correction [a]. Hence, the standard deviation is expressed by the squared distance from the image center. Further, since many pixels are used to extract a line segment, the longer line segments should be more accurate. Therefore, based on the relation between the number of samples and errors in a Gaussian distribution, standard deviation is assumed to be inversely proportional to the square root of line segment length l. Based on these assumptions, covariance matrix $\mathbf{R}_l$ of the errors of the end points on images is expressed using the following diagonal matrix, where c_1 and c_2 are constants.

$$\mathbf{R}_l = \frac{\mathrm{diag}\left[\left(c_1 n_u^2 + c_2\right)^2, \left(c_1 n_v^2 + c_2\right)^2\right]}{l} \quad \ldots \quad (6)$$

In addition, the covariance matrix $\mathbf{R}_q$ in the vehicle coordinate system obtained after bird's-eye view transformation is expressed by the following formula in the first order of Jacobian $\partial\mathbf{b}/\partial\mathbf{n}$.

$$\mathbf{R}_q = \frac{\partial \mathbf{b}}{\partial \mathbf{n}} \mathbf{R}_l \frac{\partial \mathbf{b}}{\partial \mathbf{n}}^T \quad \ldots \quad (7)$$

Figure 5 shows an example of the bird's-eye view transformation of the line segment shown in **Fig. 4**. In the figure, the black lines are the map lines, and the blue lines are the transformed line segments; the red eclipses are error eclipses of the end points of the line segments. The figure shows that there are small errors in the case of the line segments on the near side and in the center of the width direction, and there are large errors in the case of those on the far side. In our system, line segments with large covariance are threshold-processed so that they are not used in the correction of vehicle position. Thus, line segments with large errors in the peripheral area of an image or small line segments are removed on a priority basis.

3.5. Association Between Map Lines and Image Lines

For correcting the vehicle position estimation, the association between the line segments on images $\mathbf{q}_s$ and $\mathbf{q}_e$ and those on maps $\mathbf{m}_s$ and $\mathbf{m}_e$ is studied. The end points of the line segments in the vehicle coordinate system are transformed to points in the world coordinate system through

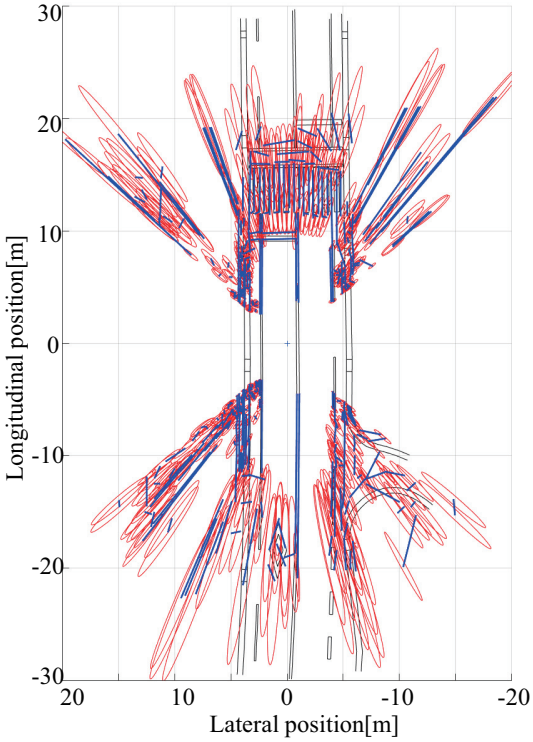


Fig. 5. Bird's-eye view transformation of lines and observation errors.

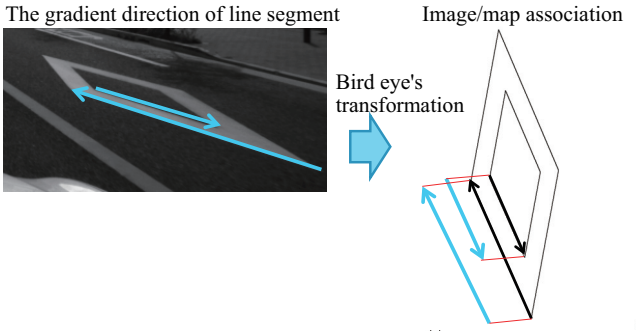


Fig. 6. Association with focus on gradient direction.

by estimating $\bar{\mathbf{x}}^t$ using the vehicle movement model. The transformed points can be then associated with the map line segments located nearby. However, since a white road line has two edges that are both detected as line segments, association with nearby line segments could lead to wrong association with the wrong edge. Hence, the thickness of a color on both sides of a line segment is used to calculate the direction of the color gradient, which is then matched to the direction of the map line segment (**Fig. 6**). This can largely reduce erroneous association. To summarize: (1) the projection distance from an image line segment to a map line segment should be smaller than a threshold value. (2) The angle difference should be smaller than a threshold value. (3) The direction of the gradient of the map line segment should be the same as that of the image line segment. Since a line is sometimes blurred and detected as a set of small lines, a single image line segment

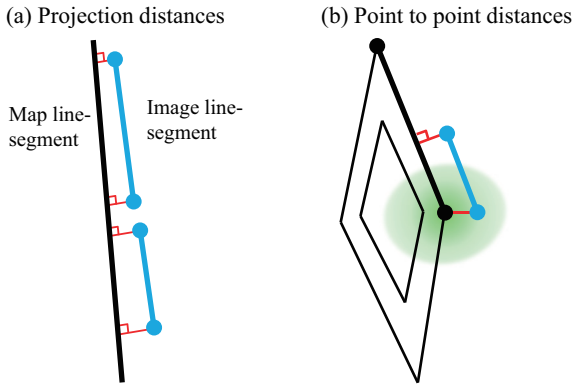


Fig. 7. Errors between line segments on map and on image.

is associated with multiple map line segments that satisfy the association conditions.

The detected line segments of objects such as blurred white lines and cracks on the road do not correspond to any map line segments. If these are directly associated with the map line segments, they may lead to incorrect vehicle localization. Hence, vehicle localization correction, which will be described in the next section, was conducted based on bisquare weighting, a robust estimation; the correction was performed for the projection distance and angle in the association of the line segments. This prevents the influence of incorrect associations if the percentage of correct associations is sufficiently larger than that of incorrect associations.

3.6. Vehicle Position Correction

Errors between the line segments on the map $\mathbf{m}_s$ and $\mathbf{m}_e$ obtained through association and those on the image $\mathbf{q}_s$ and $\mathbf{q}_e$ are defined to correct the vehicle position. The orientation of each line segment is rather accurate because it is calculated from many pixels, but the positions of the end points of the line segments are not accurate. This is because a line segment often terminates randomly. In particular, when a long line such as center line is partly observed in an image, it is not clear which part of the center line is observed. Moreover, when a curve such as the boundary of a sidewalk is observed, it is detected as multiple broken lines. Therefore, two types of errors were defined (**Fig. 7**).

The end points of a line segment in the image were projected onto the corresponding line segment in the map, and the projection distance was used to define basic errors (**Fig. 7(a)**). This projection was made mainly for white road lines, curb stones on the right and left sides of the vehicle, and stop lines extending in a direction perpendicular to vehicle motion. Using this projection method, even a part of a long line or partly broken lines can be used for correcting vehicle localization. When the equations of the straight lines calculated based on the map line segment are $a_x x_w + a_y y_w + a_c = 0$ and $\sqrt{a_x^2 + a_y^2} = 1$, the projection error δ_p of the end point $\mathbf{q}$ of the map line segment is expressed by the following formula.

$$\delta_p = \begin{bmatrix} a_x & a_y & 0 & a_c \end{bmatrix} \begin{bmatrix} \mathbf{R}_s^t \mathbf{R}_{vw}^t \mathbf{q} + \mathbf{t}_{vw}^t \\ 1 \end{bmatrix} \quad . \quad (8)$$

The observation error can be calculated as R_p by the abovementioned covariance matrix $\mathbf{R}_q$ in the direction perpendicular to the map line segment.

$$R_p = \begin{bmatrix} a_x & a_y \end{bmatrix} \mathbf{R}_q \begin{bmatrix} a_x & a_y \end{bmatrix}^T \quad . \quad . \quad . \quad . \quad . \quad (9)$$

Next, if the vehicle position estimation error is small, the end point of a line segment on the map and that of a line segment on the image should be close to each other. Therefore, if the distance between these end points was smaller than a certain threshold, the distance was also included in the error (**Fig. 7(b)**). Then, from the dashed white lines of edge points of sidewalks, the vehicle position correction in the forward direction can be estimated. The error of the end point is defined by the following formula where $\mathbf{R}_q$ is used as the observation error.

$$\boldsymbol{\delta}_q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \left(\mathbf{q} - \mathbf{R}_{vw}^{tT} \mathbf{R}_s^{tT} \left(\mathbf{m} - \mathbf{t}_{vw}^t \right) \right) \quad . \quad (10)$$

Based on these definitions, an error vector $\boldsymbol{\delta} = [\delta_{p,1}, \delta_{p,2}, \ldots, \boldsymbol{\delta}_{q,1}^T, \boldsymbol{\delta}_{q,2}^T, \ldots]^T$ containing the errors obtained from the association with multiple line segments was defined, and the vehicle position $\mathbf{x}^t$ was corrected with the extended Kalman filter.

Fig. 8. The test vehicle and positions of the cameras.

4. Evaluation Results

Evaluation tests were conducted in cities using a test vehicle with cameras mounted on it. The results on the accuracy and the calculation speed of vehicle localization are reported in this section.

4.1. Evaluation Environment

Two industrial-use cameras (Lumenera Lm225) were mounted on the test vehicle, one on the front and another on the rear side (**Fig. 8**), to collect driving data. Images of size 1088×2048 were acquired every 100 ms, and they were reduced to size 544×1024. Since the image processing area was restricted to the road area in each image, the actual processing area of the front camera was 229×1024 and that of the rear camera was 247×1024.

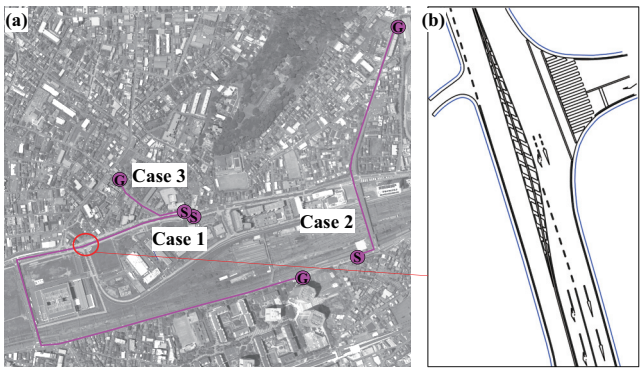


Fig. 9. Evaluation test route and the line map.

The rear wheel speed data were obtained through CAN from the sensors already mounted on the vehicle. The wheel rotation period was about 33 ms, and vehicle motion estimation was continued until the next image was acquired using the movement model described in Section 3.2. The actual latitude and longitude of the vehicle were obtained from RTK-GPS (Novatel OEM615), and these data transformed to the map coordinate system, i.e., a plane rectangular coordinate system. Since the present study focused on updating vehicle position, the true position was used as the initial position of the vehicle. Further, the present study did not focus on vehicle localization at night or under bad weather conditions and the test driving data were collected around noon on a sunny day in March.

Three test routes (Cases 1 to 3) were chosen near Shin-Kawasaki Station where RTK correction was effective (**Fig. 9(a)**). Each route included a turn at a crossing. In Case 1, the route was 1.4 km long and included a land bridge and a slope of about 3° gradient. In Case 2, the vehicle made a left turn at a crossing and then drove along a gentle curve. In Case 3, the vehicle made a right turn and then drove into a narrow city street. Accurate line segment maps were prepared for all the cases (**Fig. 9(b)**) by a specialized company. For creating the maps, measurements were made using MMS. Reference points were separately measured for verification, and comparison of positions showed that the errors in the created maps were within 0.1 m. The maps contained line segments representing white lines, yellow lines, road signs painted on the roads, and other road features such as curb stones, road gutters, and rain water inlets.

For the comparison of precision, a vehicle localization system using laser scanner reflection intensity, described in [12], was used as a reference. In [12], the intensity of the reflection from a point group, obtained from a laser scanner, is compared with the information obtained from a map through normalized correlation, and vehicle position is estimated with a particle filter. In this reference system, raster map of laser reflection intensity was used, and in the present method, line segment maps are used. Hence, the method in [12] cannot be used directly in the present method. Therefore, the reflection intensity levels of the areas such as white or yellow lines where the reflection is clearly stronger than on the road surface were set as 1 and that of the road surface was set as 0 in order to calculate the normalized correlation. A laser scanner Velodyne HDL32e was mounted on the roof of the test vehicle. For achieving an appropriate S/N ratio of the reflection intensity, the detection area within a radius of 15 m from the vehicle was used for vehicle localization. In this area, the road surface extends in all directions; there are about 17 layers in the vertical direction, and this can help realize sufficiently high resolution.

4.2. Accuracy Evaluation of Vehicle Localization

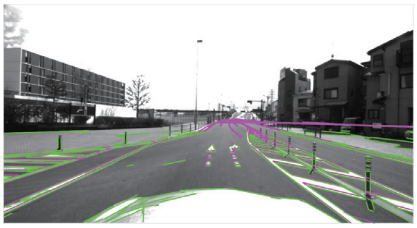
For comparing vehicle localization results and true positions, the errors are classified as those along the motion direction and those along the width direction. For applications such as autonomous vehicle driving or lane keeping, the errors in the width direction should be minimized to maintain the vehicle on the given lane. **Table 1** lists the average errors for each of the cases with front and rear cameras or with only a front camera. The average error in the width direction with two cameras was 0.1 m or smaller. The survey of the width direction errors for driving in cities in Reference [15] indicates that the total error, including that caused by control and by vehicle localization, should be 0.2 m or less. Further, Reference [23] suggests that with regard autonomous parking operations, the error should be 0.2 m or less. The average error in the present method satisfies this requirement. Even when the error in the driving direction is included, the total average error is within 1 m. Although direct comparison is difficult because of different evaluation environments, different sensors, and different maps, one can still say that the present method achieves sufficiently high accuracy – as high as or even higher than that of vehicle localization in the studies in Reference [10].

Some examples of vehicle localization are shown in **Fig. 10**. The green lines are the line segments extracted from the images. The magenta lines are the segments from the map projected onto the images based on the estimated vehicle position. Although calibration problems persist, the maps and images overlap with each other if the vehicle position is correctly estimated. **Fig. 10(a)** presents a scene with complicated white lines and road signs, and **Fig. 10(b)** shows a scene where the vehicle moves on a land bridge with 3° slope. **Fig. 10(c)** shows a scene where the vehicle is passing by a parked car; **Fig. 10(d)**, a scene where the vehicle moves on a narrow street; **Fig. 10(e)**, a scene that will be explained later. As all these figures show, the vehicle position was correctly estimated.

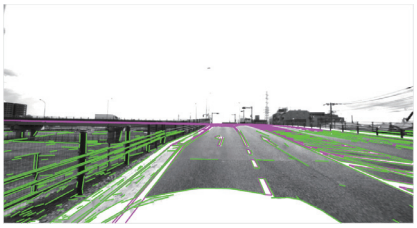
Figure 11 shows the error distribution in Case 1. In this figure, the errors are plotted for each instant of image acquisition. The errors are distributed on and around 0. The errors in the driving direction (longitudinal position errors) are larger than those in the width direction (lateral position errors) because there were no clues for vehicle localization when driving on a long straight road. When there are no clues on the map, or in other words, when there is no change in landscape, large errors will not

Table 1. Average errors in the estimated vehicle position.

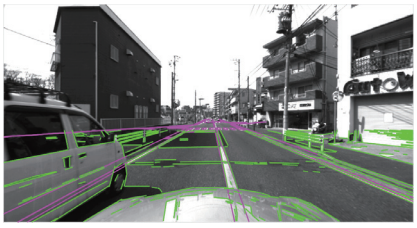
Evaluation Paths	Distances [m]	Methods	Lateral position error [m]	Longitudinal position error [m]
Case 1	1381	Proposed (Front + Rear)	0.049	0.21
		Proposed (Front)	0.087	0.19
		Laser based	N/A	N/A
Case 2	534	Proposed (Front + Rear)	0.058	0.63
		Proposed (Front)	0.083	0.69
		Laser based	0.059	1.10
Case 3	148	Proposed (Front + Rear)	0.096	0.23
		Proposed (Front)	0.13	0.17
		Laser based	0.053	0.16



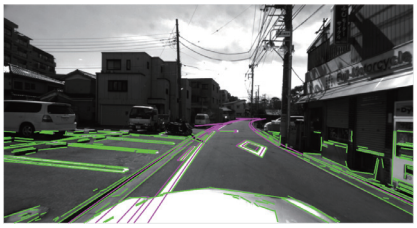
(a)



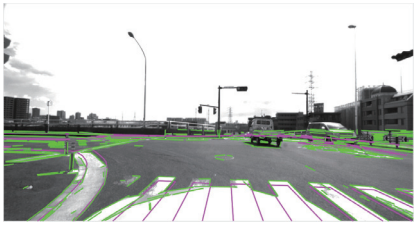
(b)



(c)



(d)



(e)

Fig. 10. The map projected onto the image.

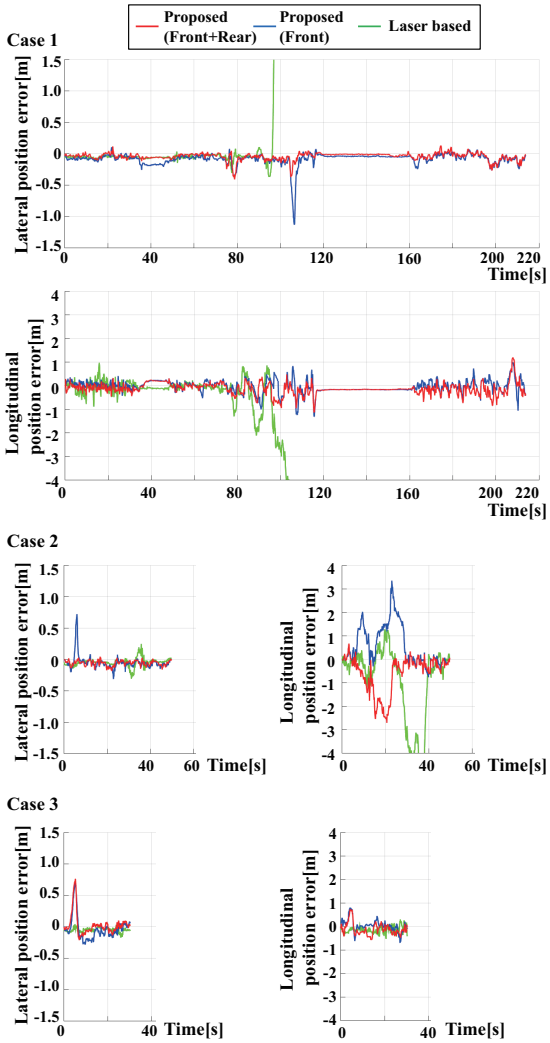


Fig. 11. Estimation error distributions.

pose a big problem in actual applications. On the other hand, the errors in the width direction were large in some cases when only one camera was used; **Fig. 10(e)** shows an example where the error was caused when the vehicle moved up a land bridge and made a turn at a crossing with fewer clues. Errors attributed to odometry tend to be large

when the vehicle makes a turn. If only a limited number of clues are available, for example, because the crosswalk paint is blurred, the error tends to be large. However, the use of more than one camera was found to be effective because in this case, the rear camera captured the crosswalk and prevented large errors.

The comparison with the laser method (**Table 1** and **Fig. 11**) is discussed here. In Case 3, the present method and the laser method had almost the same accuracy. In Case 1, the longitudinal position error became large when the vehicle drove on a straight road near the land bridge, and hence, subsequent vehicle localization became difficult. Further, in Case 2, the longitudinal position error was large when the vehicle drove on a gentle curve. Because the present method can utilize clues that are independent of laser reflection intensity, such as rain water inlets, street gutters, and the end points of dashed white lines, the longitudinal position errors can be corrected. On the other hand, the combination of the method reported in Reference [12] and the line segment maps did not provide sufficient clues for the front side, thereby causing large errors.

Line segment detection not only involves finding clues of white road lines but also those of other features. To examine the effect of the detection of other features, the line segments of road structures such as curb stones, street gutters, and rain water inlets were eliminated from the line segment maps, and the present method was applied to the maps that contained only road paints such as white lines, yellow lines, and road signs. This additional experiment was conducted with a vehicle with front and rear cameras. The result shows that in Case 2, the lateral position error was 0.065 m, and the longitudinal position error was 0.45 m; these were almost the same as those in the case of the original maps. However, there was a large deviation in vehicle position, and further estimation became difficult when the vehicle took a turn at a crossing in Case 1 and when the vehicle moved along a narrow street in Case 3. It was thus shown that clues such as curb stones could contribute to the improvement of the robustness.

The above results shows that the present method could meet the lateral position error criterion of 0.2 m or less, which is necessary to make commercial products, and could thus realize highly accurate vehicle localization. However, vehicle localization could be difficult in some situations. First, on a road with fewer clues such as white lines or curb stones, vehicle localization with this method will be difficult. Similarly, on a road with many parked cars that cover clues, the present method will be ineffective. In these situations, association between the map line segments and image line segments cannot be obtained, and the method has to rely on the odometry-based movement model. If these situation persist, vehicle localization errors will accumulate and exceed the criterion of the association between the images and the maps; such association may require initialization by GPS. As in the case of high-precision GPS, the present method cannot be applied in all environments. If the method is used for safety-related applications such as autonomous vehicle driving, numerous methods should be applied to compensate each other's disadvantages. Nighttime, backlight, and rain are also problematic factors for cameras, and these factors will be considered as future problems. Further, the line segments of other moving cars or stopped cars, which are not present in the given maps, are removed by threshold processing and robust estimation; however, direct detection may stabilize system operation, and hence improvement can still be achieved in future. It will be also necessary to suppress error peaks by examining individual cases.

Table 2. Average calculation time of vehicle localization.

	Calculation time [ms]
Line segment extraction (Front camera)	13
Line segment extraction (Rear camera)	15
Localization	12
Total	41

4.3. Evaluation of Calculation Time

The average calculation time of the present method is listed in **Table 2**. For vehicle localization, a PC with Intel core i7, NVIDIA GeForce GTX750, and Windows 8 was used, and only CPU was used for the calculation. The calculation time did not include the drawing time. The simulation software was implemented with MATLAB and C. Since images were acquired every 100 ms, real-time calculation was sufficiently feasible.

5. Summary

This paper proposes a method of estimating vehicle position accurately by matching line segment maps and line segment features extracted from multi-camera images. Clues for vehicle localization, such as white road lines, yellow road lines, road signs, and curb stones were treated in an integrated manner by extracting their line segment features. The detected line segments were transformed into the vehicle coordinate system by bird's-eye view transformation in order to realize processing independent of the camera configuration. The extended Kalman filter was applied after a detailed examination of the line segment observation errors to realize real-time vehicle localization. The accuracy of the localization was evaluated by conducting driving experiments in cities, and the results indicated that the localization was highly reliable with sub-meter accuracy.

References:
[1] J. Ziegler, P. Bender, M. Schreiber et al., "Making bertha drive? An autonomous journey on a historic route," IEEE Intelligent Transportation Systems Magazine, Vol.6, pp. 8-20, 2014.
[2] J. McCall and M. Trivedi, "Video-based lane estimation and tracking for driver assistance: Survey, System, and Evaluation," IEEE Trans. on Intelligent Transportation Systems, Vol.7, No.1, pp. 20-37, 2006.
[3] B. Wu, T. Lee, H. Chang et al., "GPS navigation based autonomous driving system design for intelligent vehicles," IEEE Int. Conf. on Systems, Man and Cybernetics, pp. 3294-3299, 2007.
[4] M. Noda, T. Takahashi, D. Deguchi et al., "Vehicle ego-localization by matching in-vehicle camera images to an aerial image," Asian Conf. on Computer Vision 2010 Workshops – Computer Vision, pp. 163-173, 2011.
[5] D. Wong, D. Deguchi, I. Ide et al., "Single camera vehicle localization using SURF scale and dynamic time warping," Proc. of the IEEE Intelligent Vehicles Symposium, pp. 681-686, 2014.
[6] H. Lategahn, M. Schreiber, J. Ziegler et al., "Urban localization with camera and inertial measurement unit," Proc. of the IEEE Intelligent Vehicles Symposium, pp. 719-724, 2013.
[7] H. Yu, H. Hsieh, K. Tasi et al., "Visual Localization for Mobile Robots Based on Composite Map," J. of Robotics and Mechatronics, Vol.25, No.1, pp. 25-37, 2013.
[8] D. Zhang, R. Kurazume, Y. Iwashita et al., "Robust Global Localization Using Laser Reflectivity," J. of Robotics and Mechatronics, Vol.25, No.1, pp. 38-52, 2013.
[9] M. Schreiber, C. Knoppel, and U. Franke, "LaneLoc: Lane marking based localization using highly accurate maps," Proc. of the IEEE Intelligent Vehicles Symposium, pp. 449-454, 2013.
[10] S. Nedevschi, V. Popescu, R. Danescu et al., "Accurate Ego-Vehicle Global Localization at Intersections Through Alignment of Visual Data With Digital Map," IEEE Trans. on Intelligent Transportation Systems, Vol.14, No.2, pp. 673-687, 2013.
[11] F. Chausse, J. Laneurit, and R. Chapuis, "Vehicle localization on a digital map using particles filtering," Proc. of the IEEE Intelligent Vehicles Symposium, pp. 243-248, 2005.
[12] J. Levinson, M. Montemerlo, and S. Thrun, "Map-Based Precision Vehicle Localization in Urban Environments," Robotics: Science and Systems, 2007.
[13] N. Mattern, R. Schubert, and G. Wanielik, "High-accurate vehicle localization using digital maps and coherency images," Proc. of the Intelligent Vehicles Symposium, pp. 462-469, 2010.
[14] Y. Yu, H. Zhao, F. Davoine et al., "Monocular visual localization using road structural features," Proc. of the IEEE Intelligent Vehicles Symposium, pp. 693-699, 2014.
[15] J. Ziegler, H. Lategahn, M. Schreiber et al., "Video based localization for Bertha," Proc. of the IEEE Intelligent Vehicles Symposium, pp. 1231-1238, 2014.
[16] I. Puente, H. Gonzalez-Jorge, J. Martinez-Sanchez et al., "Review of mobile mapping and surveying technologies," Measurement, Vol.46, No.7, pp. 2127-2145, 2013.
[17] J. Ziegler, P. Bender, and T. Dang, "Trajectory planning for Bertha – A local, continuous method," Proc. of the IEEE Intelligent Vehicles Symposium, pp. 450-457, 2014.
[18] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," ACM Int. Symposium on Mixed and Augmented Reality, pp. 225-234, 2007.
[19] R. Pless, "Using many cameras as one," IEEE Conf. on Computer Vision and Pattern Recognition, Vol.2, pp. 587-593, 2003.
[20] S. Thrun, W. Burgard, and D. Fox, "Probabilistic robotics," MIT press, 2005.
[21] R. Gioi, J. Jakubowicz, M. Morel et al., "LSD: A fast line segment detector with a false detection control," IEEE Trans. on Pattern Analysis & Machine Intelligence, Vol.32, No.4, pp. 722-732, 2008.
[22] K. Hirose and H. Saito, "Real-time Line-based SLAM for AR," The 3rd Int. Workshop on Benchmark Test Schemes for AR/MR Geometric Registration and Tracking Method, 2012.
[23] H. Masuda, M. Mizutani, and M, Kimura, "ITS Application: First Step in Automated Cruise System," Toshiba review, Vol.55, No.11, pp. 23-26, 2000 (in Japanese).

Supporting Online Materials:
[a] J. Bouguet, "Camera Calibration Toolbox for MATLAB." http://www.vision.caltech.edu/bouguetj/calib_doc/ [Accessed June 16, 2013]

Name:
Kosuke Hara

Affiliation:
Chief Engineer, Denso IT Laboratory, Inc.

Address:
CROSSTOWER 28F, 2-15-1 Shibuya, Shibuya-ku, Tokyo

Brief Biographical History:
2002 Graduated with M.S. degree from Tokyo University of Agriculture and Technology
2004- Joined Denso IT Laboratory
2014- Doctor Course Student, Keio University

Main Works:
• "Detection of unusual human behavior in intelligent house," Proc. of the 2002 12th IEEE Workshop on Neural Networks for Signal Processing, pp. 697-706, 2002.

Name:
Hideo Saito

Affiliation:
Professor, Department of Information and Computer Science, Keio University

Address:
3-14-1 Hiyoshi, Kohoku-ku, Yokohama 223-8522, Japan

Brief Biographical History:
1992- Faculty of Science and Technology, Keio University
1997-1999 Visiting Scientist, Robotics Institute, Carnegie Mellon University
2006- Professor, Department of Information and Computer Science, Keio University

Main Works:
• H. Saito, S. Baba, and T. Kanade, "Appearance-based virtual view generation from multicamera videos captured in the 3-D room," IEEE Trans. on Multimedia, Vol.5, No.3, pp. 303-316, 2003.
• H. Uchiyama, H. Saito, M. Servieres, and G. Moreau, "Camera tracking by online learning of keypoint arrangements using LLAH in augmented reality applications," Virtual reality, Vol.15, No.2-3, pp. 109-117, 2011.
• N. Inamoto and H. Saito, "Virtual viewpoint replay for a soccer match by view interpolation from multiple cameras," IEEE Trans. on Multimedia, Vol.9, No.6, pp. 1155-1166, 2007.

Membership in Academic Societies:
• The Institute of Electrical and Electronics Engineers (IEEE), Senior Member
• The Virtual Reality Society of Japan (VRSJ), Fellow
• The Institute of Electronics, Information and Communication Engineers (IEICE), Senior Member
• Information Processing Society of Japan (IPSJ)