Estimation of Ego-Vehicle’s Position Based on Image Registration

Keisuke Kazama 1)  Toshiki Kawakatsu 1)  Yasuhiro Akagi 1)  Hiroshi Mouri 1)
1) The Tokyo University of Agriculture and Technology, Department of Mechanical Systems Engineering
2-24-16 Naka-cho, Koganei-shi, Tokyo, 184-8588, Japan

Received on November 17, 2017

ABSTRACT: This paper describes an estimation method of ego-vehicle’s position using a 2D map. The general approach for localization is to match the white lines between the map and the real world. However, such approaches suffer from changing the environmental conditions and painting clearance. If considerable changes occur, the false-detection rate increases due to noises and miss-detected lines. In order to solve this issue, a localization method based on the holistic road area detection is proposed. First, the road area is extracted from a predefined 2D boundary map. Next, the real world road area is detected using LiDAR and converted into the binary image plane. Finally, an image registration technique is applied to calculate the overall matching score of the road area between the map and LiDAR images. The proposed method has provided an accurate estimation against environmental changes with low-cost calculation based on the simulation results. In addition, the validation of the proposed method in the real world has performed less than 0.2 m for the estimation error.

KEY WORDS: Safety, Active Safety, Autonomous Driving, Localization, Image Processing, Image Registration [C1]

1. Introduction

Localization in autonomous driving is used to estimate the position and travelling direction of the vehicle with respect to global map. The minimum level of estimation accuracy required for autonomous driving is such that the driver can recognize the travelling lane. Therefore, many studies use a 3D map as the global map because it enables highly accurate position estimation in a complex environment where many obstacles exist (1-3). Yoneda et al. proposed 3D point cloud matching that uses the ICP algorithm and achieved a position estimation error of 0.3 m in real time (1). However, this method requires correspondence between observation points in the 3D space and the 3D map, so the processing time is imposed on square observation points. Furthermore, it is expensive cost to create and maintain the 3D map, and a huge memory allocation is required to carry the map information onboard.

2D Google maps are widespread, readily available and can easily be used as onboard maps. These maps are composed of boundaries of roads, pavements, and buildings. In order to estimate the vehicle position using a 2D boundary map, various methods are used to detect road boundaries (e.g., white lines and curbs) in the driving environment and align them to the boundaries in the 2D map.

Many studies have focused on autonomous driving on highways because detecting lane lines is relatively easy (4-8). Yamaguchi et al. proposed a white line detecting method that does not overlook any edge characteristic points while suppressing the noise to below the threshold of the Sobel filter by emphasizing the edge of a large area while smoothing the image locally (4). Muramatsu et al. performed edge extraction using a Krisch operator and the least mean error method by combining them with heuristic rules to detect white lines and preceding vehicles (5). Differential processing is required to detect boundaries such as white lines on the road. However, if it is possible to detect even slight boundary lines, it is more susceptible to noise and false-detection increases. By contrast, in order to suppress the influence of noise to a small level, miss-detection increases. Estimation accuracy of the vehicle position decreases greatly if false-detection or miss-detection of the white line occurs. Various measures have been examined for detecting white lines (9-11). However, no fundamental solution has yet been found, and estimating ego-vehicle’s position using white line remains difficult.

In the augmented reality research field of image processing, it has been clarified that matching accuracy improves by using mesh information linking point clouds rather than simple matching of point clouds (12-15). False-detection or miss-detection of white lines can be avoided fundamentally by using only the point on the road from the observation points as the “road area” instead of detecting the white lines from the observation points. In other words, information for estimating ego-vehicle’s position concentrates on the boundary, and highly accurate estimation is possible if the boundary can be detected with certainty. However, it is difficult to reliably detect line information. On the other hand, although the information value held by each point in the road area is small, even if a little miss-detection occurs, the influence is small. Therefore, this study proposes an estimation of ego-vehicle’s position method based on the holistic road area rather than white lines to estimate the vehicle position and heading with high accuracy while reducing the processing time.

The road area is extracted and converted into a road image from the 2D boundary map. The vehicle position and heading angle are estimated using Lucas-Kanade algorithm (LK). LK calculates the matching score between the map road image and the road area that is being extracted from the LiDAR image. Meanwhile, these two quantities are measured using dead reckoning based on the vehicle velocity and elapsed time. Finally, Extended Kalman Filter (EKF) is utilized to optimize the measurements of LK and dead reckoning and obtain the pose and heading angle.
We confirmed the validity of the proposed method using simulation as a basic assessment that allows to examine situations where the dead reckoning estimation is considerably wrong and the road area cannot be detected accurately because of obstacles. In addition, the performance of the proposed method has been verified by collecting real data on public roads using the experimental vehicle.

2. Vehicle Position Estimation based on Image Registration

Figure 1 depicts the processing flow of the proposed ego-vehicle’s position estimation method. First, vehicle velocity $V_t$ and yaw rate $\gamma$ obtained from the vehicle speed sensor and yaw rate sensor mounted on the vehicle are input as control data. The current vehicle position and heading are then predicted using the vehicle motion model expressed by equation (1). Here, over-line (’) indicates the predicted value and hat (^) indicates the estimated value after updating the measurement. $\Delta t$ is the sampling time (sec), and suffix $t$ is the current step.

$$
\begin{bmatrix}
\dot{X}_t \\
\dot{Y}_t \\
\dot{\theta}_t
\end{bmatrix}
= 
\begin{bmatrix}
V_t \cos \hat{\theta}_{t-1} \\
V_t \sin \hat{\theta}_{t-1} \\
\gamma_t
\end{bmatrix}
\Delta t
$$

(1)

Meantime, a road area detection method based on 3D LiDAR point cloud is applied. The detected area is converted into an image $I_{global}(x)$. Here, $x = (x, y)^T$ is the vector indicating the coordinates on the image, and the origin is in the upper left corner of the image. The road area image $M(x)$ is prepared by extracting the road area from the map of the travelling environment. The subject image $I(x)$ to be used for image registration is extracted from image $I_{global}(x)$ based on the vehicle heading. Details will be described in 2.1. Based on the vehicle position and its heading, reference image $T(x)$ is extracted as the standard of registration. Translation and Rotation parameters of the vehicle, $p = (t_x, t_y, r)^T$, are obtained from $I(x)$ and $T(x)$ using the image registration method. The vehicle position and its heading are updated using the extended Kalman filter in equation (2). Translation and Rotation parameters $p$ are used as the observed values.

$$
\begin{bmatrix}
\hat{X}_t \\
\hat{Y}_t \\
\hat{\theta}_t
\end{bmatrix}
= 
\begin{bmatrix}
\hat{X}_t + t_x \\
\hat{Y}_t + t_y \\
\hat{\theta}_t + r
\end{bmatrix}
+ 
K_{e}
\begin{bmatrix}
\hat{X}_t - X_t \cos \hat{\theta}_t - Y_t \sin \hat{\theta}_t \\
\hat{Y}_t - X_t \sin \hat{\theta}_t + Y_t \cos \hat{\theta}_t \\
\hat{\theta}_t - \gamma_t
\end{bmatrix}
$$

(2)

The environment around the vehicle is scanned using LiDAR which provides a 3D point cloud. The point clouds are accumulated continually to enhance the density representation with respect to the vehicle velocity. Histogram of Oriented Gradients (HOG) of the incline magnitude and direction of the distance data is then calculated. Accordingly, Support Vector Machine (SVM) is applied to detect the road area as detailed in reference (18). The classification results using the KITTI database are shown in Table 1. For comparative targets, we chose UNV which identifies the highest performance in the database and PGM-ARS (17) which uses machine learning as well as us. The proposed method achieves accuracy comparable to the UNV method except multiple marked road. Recall performance is not high. This is a factor of impairing the performance of position estimation. But, if it is possible to acquire an important intersection shape or curve shape in order to estimate the position, it will be shown later that it is not affected much. In addition, although the performance as a classifier has been lowered, it can be expressed that it is carrying out the classification of safety classification not to classify places that are not roads as roads. On the other hand, the reason why the accuracy deteriorated in multiple marked road has not yet been clarified. The current ability is applicable to roads of one lane on either side. From now on, we aim to improve precision by repeatedly learning on a wide road.

2.1. Road area detection and, Road area image preparation

The top left corner is considered as the reference of the absolute coordinate system regardless to the vehicle heading direction. Based on the vehicle center of gravity, the detected road area of 60x60 meter is converted into the image domain and represented by an image $I_{global}(x)$ of $1024 \times 1024$ pixels, i.e., the road area is indicated by 255 pixel values. Figure 2a depicts a 3D LiDAR point cloud of scanning a road junction. The color range changes from blueeness to redness to represent the lowest and highest detected objects with respect to the vehicle plane, i.e., green-yellow points indicate zero meter-level. The road area is classified using the HOG characteristics and SVM. The classified area is then converted into the global image $I_{global}(x)$ in white pixels as shown in Fig. 2b.

Finally, the global image is resized to a subject image $I(x)$ of 512x512 pixels to be used for image registration as illustrated in Fig. 2c. As long as the LiDAR point clouds are accumulated with respect to the vehicle heading angle, the density of road representation will be higher in the previous frames. Therefore, 8 meter ahead are only considered to set the region of interest (ROI) in order to extract $I(x)$ as can be observed in Figs. 2b and 2c.
2.2. Method for Extracting the Road Area from Map Information

The map information used in this study is a 2D boundary map that provides by the Geospatial Information Authority of Japan. This map is composed of end-point information of road boundaries as illustrated in Fig. 3a. Global map image $M(x)$ is prepared in size of 512×512 pixels as demonstrated in Fig. 3c. The map information $M(x)$ depicts the area around the Koganei campus of the Tokyo University of Agriculture and Technology (image size 15702×5632 pixels). Based on the vehicle position and heading angle that predicted using equation (1), reference image $I(x)$ is extracted from the map image $M(x)$ in size of 512×512 pixels as demonstrated in Fig. 3c.

2.3. Image registration

Image registration is applied between a subject $I(x)$ and reference $T(x)$ images to minimize the error of translation, rotation, scale, expansion and so forth. This method was proposed in the Lucas-Kanade optical flow algorithm (18-19) and has been widely used in computer vision applications. In this paper, the original Lucas-Kanade algorithm is used. Squared sum error $E$ between the two images is expressed in equation (3).

\[
E = \sum_x (I(x, p) - T(x))^2
\]

As the transformation matrix $W(x, p)$ is nonlinear, translation and rotation parameters $p$ cannot be obtained analytically. Therefore, the solution is optimized by defining the rate of change $\Delta p$ of the parameters and performing iterative computation. Square sum error $E$ is rewritten as expressed by equation (3)’:

\[
E = \sum_x (I(x, p + \Delta p) - T(x))^2
\]

By considering Taylor derivation of the first term of equation (3) and ignoring the quadratic and higher-degree terms, Eq. (5) is obtained.

\[
I(x, p + \Delta p) \approx I(x, p) + \left( \frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial p} \Delta p \right) \approx I(x, p) + \frac{\partial I}{\partial x} \Delta p
\]

Accordingly, the square sum error $E$ can be expressed by equation (6).

\[
E = \sum_x \left( I(x, p) + \frac{\partial I}{\partial x} \Delta p - T(x) \right)^2
\]

By differentiating this equation with respect to $\Delta p$, the value of $\Delta p$ that minimizes the square sum error $E$ can be calculated by equation (7).

\[
\frac{dE}{d\Delta p} = 2\sum_x \left[ \frac{\partial I}{\partial x} \frac{\partial I}{\partial p} \right] \left( I(x, p) + \frac{\partial I}{\partial x} \Delta p - T(x) \right) = 0
\]

\[
\Delta p = -\left( \sum_x \left[ \frac{\partial I}{\partial x} \frac{\partial I}{\partial p} \right] \right)^{-1} \sum_x \left[ \frac{\partial I}{\partial x} \frac{\partial I}{\partial p} \right] (T(x) - I(x, p))
\]

If the rate of change of square sum error $\Delta E = E(x, p + \Delta p) - E(x, p)$ is sufficiently small, translation and rotation parameters $p$ are determined. Otherwise, the calculation is repeated assuming that $p = p + \Delta p$.

Since the calculated parameter $p$ are the translation and rotation parameters at the center of the image, it is converted into the vehicle coordinate system using equation (8):

\[
p = \begin{bmatrix} t_x \\ t_y \\ r_x \\ r_y \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \\ r_x \\ r_y \end{bmatrix} + \begin{bmatrix} t_x \cos(r_x + r_y) - \cos(r_y) \\ t_y \sin(r_x + r_y) - \sin(r_y) \\ 0 \end{bmatrix}
\]

Using the new parameter $p$ as the observed value of the extended Kalman filter, the predicted values of the vehicle position and its heading angle are corrected based on equation (2).

Figure 4 depicts the results of applying image registration between subject image $I(x)$ in Fig. 3c and reference image $T(x)$ in Fig. 2c. Figure 4b presents the error image $(I(x) - T(x))$, i.e., the black areas refer to $I(x)$, the white areas belong to $T(x)$ and the gray areas indicate zero difference. The area that enclosed by an orange rectangle in Fig. 4b represents the translation error between the two images. After applying image registration, this area has been converted to gray in Fig. 4c to indicate that the difference between the images has been detected and the translation and rotation error has been minimized.
3. Confirmation of the Principle by Simulation

3.1. Simulation condition

A simulation was prepared to confirm the principle of the proposed method under two conditions: (1) the road area cannot be detected accurately and (2) the vehicle behavior cannot be understood because of sensor redundancy. Figure 5a illustrates the driving environment of 40×40 m square area. The blue line indicates the driving trajectory of the vehicle, where the road is bordered by curbstones with a height of 0.1 m. This environment is converted into the image domain as shown in Fig. 5b. In order to simulate some critical driving situations, five obstacles that represent parked vehicles are placed on the target course as demonstrated in Fig. 5c. Each obstacle is 2.6 m long and 1.6 m wide. Because the vehicle must travel and avoid the obstacles, it meanders through the course.

During this simulation, vehicle velocity $V$ was kept constant at 5 km/h and yaw angular velocity $\gamma$ was calculated based on the two-wheel model by estimating the steering angle in advance and sampling time $\Delta t$ was set to 0.1 sec.

3.2. Simulation result

The vehicle pose and heading angle were estimated by applying Gaussian function noise to the control inputs of velocity $V$ and yaw angular velocity $\gamma$, i.e., the noise at vehicle velocity $V$ is 0, standard deviation is 0.1 m/s, yaw angular velocity $\gamma$ is 0, and the corresponding standard deviation is 0.1 rad/s. Figure 6a shows the actual travelling trajectory of the vehicle denoted by blue line whereas the predicted trajectory is represented by green dashed line based on vehicle velocity $V$ and yaw angular velocity $\gamma$ (inertial navigation: INS). INS may predict a trajectory that deviates largely from the actual travelling trajectory because of noise. In contrast, the red broken line (proposed method) in Fig. 6b almost completely matches the actual travelling trajectory. Here, the estimated RMS errors were longitudinal position $=0.08$m, and lateral position $=0.03$m.

The computer used for this simulation is an Intel Core i7-4600U (2.10GHz, 2 cores), which performed all processing from road area detection to image registration in an average of 0.0887sec (up to 0.0957sec). The number of iterations of image registration was about 7 on average. Results confirmed that the proposed method enables accurate vehicle pose estimation with only short processing time.

4. Performance Evaluation using real travelling data

4.1. Setup

In this section, we verify the validity of the proposed method by applying it to real travelling data. The driving environment is composed of public roads around Midori-cho, Koganei-shi, Tokyo (Fig. 7), and the vehicle travels a route of 2,000m per one lap (blue line in Fig. 7a). 80% of the environment is composed of narrow roads with no center lane line (area A); the remaining 20% is composed of one-way roads (area B). The experiment was operated at 15:00 on a weekday with a mixture of obstacles such as pedestrians, bicycles, and travelling vehicles. The average vehicle velocity excluding stops at signals and crossings was 25 km/h.

A Prius (Toyota Motor Corp., Fig. 8) was used for data measurement. A 3D LiDAR (Velodyne, HDL-32E) mounted onboard detects the road area and scans the surrounding environment at a frequency of 10 Hz. An IMU (Sumitomo Precision Products Co., Ltd.) mounted on the vehicle measures the 3D acceleration and angular velocity of the vehicle and performs data measurement at frequency of 100 Hz. It also measures vehicle velocity $V$ and steering angle $\theta$ at frequency of 100 Hz. A highly precise GNSS (Javad, Delta) is mounted as the target for comparing vehicle position estimation. This method uses network RTK-GNSS (provided by JENOBA Co., Ltd.), which requires no fixed base.
station, and the minimum error of location accuracy is 0.02 m when
the GNSS positioning mode is in the RTK fixed state.

Data is measured while operating the test vehicle in accordance
with the official vehicle operation and management rules of Tokyo
University of Agriculture and Technology. The person responsible
for the test, the driver, and the management assistant rode in the
test vehicle and attention was drawn as not to cause annoyance to the
surroundings.

\[ X = \] especially in the area near the coordinate of (\( -29 \) km, \( -32.8 \) km), i.e., the position deviates greatly from the road on the
map. These results confirm that the vehicle pose estimation for
autonomous driving is difficult to be achieved with only simple
map. In contrast, the proposed method (red dotted line) can
provide almost the same results in all three figures. The travelling direction almost overlaps the result of GNSS
positioning and the error in \( X \) direction is very small as well. Figure 9c shows the time series of 2.0 sec during travelling along a final
straight course. Although the lateral direction can be estimated with
an error of about 0.20 m, an error of about 2.0 m is generated in the
longitudinal direction. Since the shape of the road does not change
on a long straight road, it is directly influenced by accumulation
error of odometry. In this regard, coordinates of traffic lights and
road signs are input in advance on the map, and we are considering
whether we can improve by using target recognition. The estimated
RMS errors were longitudinal position = 0.85m, and lateral
position = 0.16m. The number of iterations of image registration
was about 16 on average.

4.2. Result

Figure 9 presents the verification results. Figure 9b presents an
enlarged view of area A, and Fig. 9c presents an enlarged view of
area B. In both areas, the GNSS positioning mode is in the RTK
fixed state; therefore, the travelling traces are highly reliable. In
Area A the vehicle turns to the left at a crossing, and in area B the
vehicle travels along a straight road. In Fig. 9a, the trajectory
indicated by the blue line is GNSS positioning information, and the
trajectory indicated by the green dashed line is the one predicted
from vehicle velocity \( V \) and yaw angular velocity \( \gamma \) (inertial
navigation: INS). The GNSS positioning result is reliable where the
GNSS positioning mode is in the RTK fixed state; however, the
positioning mode is in standalone in the area near the upper left
corner (\( X = -29.0 \) km, \( Y = -32.8 \) km). Thus, the GNSS positioning
result is outside the road on the map. In other words, once another
mode is set, GNSS positioning becomes unreliable. In addition,
position estimation by inertial navigation is affected by noise and,
especially in the area near the coordinate of (\( X = -28.6 \) km, \( Y = -33.2 \) km), i.e., the position deviates greatly from the road on the
map. These results confirm that the vehicle pose estimation for
autonomous driving is difficult to be achieved with only simple
sensing. In contrast, the proposed method (red dotted line) can
estimate the vehicle pose without deviating from the road.

Figure 9b consists of three sections; the upper section depicts
the \( X \)-\( Y \) coordinate, the middle section depicts the time history of
the \( X \) coordinate and the lower section depicts the chronological
order of the \( Y \) coordinate. Position was estimated 20 times during
a 2.0 sec left turn, and GNSS positioning and the proposed method
provide almost the same results in all three figures. Figure 9c
depicts the time series of 5.0 sec during travelling along a straight
course. The travelling direction almost overlaps the result of GNSS
positioning and the error in \( X \) direction is very small as well. Figure 9c shows the time series of 2.0 sec during travelling along a final
straight course. Although the lateral direction can be estimated with
an error of about 0.20 m, an error of about 2.0 m is generated in the
longitudinal direction. Since the shape of the road does not change
on a long straight road, it is directly influenced by accumulation
error of odometry. In this regard, coordinates of traffic lights and
road signs are input in advance on the map, and we are considering
whether we can improve by using target recognition. The estimated
RMS errors were longitudinal position = 0.85m, and lateral
position = 0.16m. The number of iterations of image registration
was about 16 on average.

For localization estimation, performance evaluation was carried out
using database of KITTI. The results are shown in Table 2. For
comparison, we chose the highest performance V-LOAM in the
database and STEAM-LEC using Lidar as we did. The database
has data from 10 km/h to 90 km/h. In the verification experiment,
it was possible to evaluate only about 25 km/h, but using the
database also makes it possible to evaluate the performance in the

![Fig.7 Validation environment](image)

![Fig.8 Vehicle for data logging](image)

![Fig.9 Validation results](image)
high speed range. Although the performance of Translation is low, it is 1.39% in the longitudinal direction and 0.80% in the lateral direction, and the influence of the error in the longitudinal direction is large. For improving the localization accuracy in the longitudinal direction, we are planning to apply methods based on road signs and road marks. It can be seen that the processing time is shorter even compared with the two methods. 

Table 2. Localization evaluation

| Method     | Translation | Rotation | Runtime | Environment |
|------------|-------------|----------|---------|-------------|
| Proposed   | 1.84%       | 0.0055 [deg/m] | 0.05sec | (77500U) @ 2.70GHz (MATLAB) |
| V-LOAM     | 0.63%       | 0.0014 [deg/m] | 0.10sec | 2 cores @ 2.5 GHz (C/C++) |
| STEAM-LEC  | 1.16%       | 0.0057 [deg/m] | 0.10sec | 1 cores @ 2.5 GHz (C/C++) |

Figure 10 shows the longitudinal and lateral RMS error in a certain driving. Blue line indicates the longitudinal error, and red one indicates the lateral error. It can be seen that the dispersion of the error in the longitudinal direction is larger than that in the lateral direction. For improving the localization accuracy in the longitudinal direction, we are planning to apply methods based on road signs and road marks.

Fig.10 Longitudinal and lateral RMS errors

5. Conclusion

We proposed a method of estimating the vehicle position that uses a 2D boundary map and a road area detection technique to solve the trade-off between accuracy and processing time.

The core idea is to use image registration for minimizing the error of translation and rotation between map and observation images. Our conclusions are as follows.

- We proposed ego-vehicle’s position and its heading estimation that uses image registration and confirmed the validity of this method.
- Some simulations of critical situations where the road area cannot be completely detected and the reference map differs from the actual environment are provided. The proposed method has performed high accurate estimation with existence of errors in the initial position or observation noise.
- We found that the proposed method operates within 0.1 sec using a common computer instead of a computer with a special CPU and parallel processing device used for 3D point-cloud matching. Thus, it was confirmed that this method requires only a small processing time.

- We performed road area detection using 3D LiDAR and confirmed that the road area can be detected accurately with presence of multiple pedestrians and parked vehicles.
- Verification using actual travelling data confirmed that the proposed method enables ego-vehicle’s position estimation with an RMS error of 0.20 m or less.

Acknowledgement

This study was conducted with the support of JSPS fellow encouragement funds 1J08100.

This paper is written based on a proceeding presented at JSAE FAST-zero’17 Meeting.

References

(1) Yoneda, K., Tehran, H., Ogawa, T., Hukuyama, N. and Mita, S. : Laser Scan Feature for Localization with Highly Precise 3-D Map, Proceedings of 2014 IEEE Intelligent Vehicles Symposium, pp.1345-1350 (2014).
(2) Deusch, H., Wiest, J., Reuter, S., Nuss, D., Fritzsche, M., and Dietmayer, K. : Mult-sensor Self-localization Based on Maximally Stable Extremal Regions, Proceedings of 2014 IEEE Intelligent Vehicles Symposium, pp.555-560 (2014).
(3) Akagi, Y., and Raksincharoensak, P.: A Vehicle trajectory Estimation for a Driving Behavior Analysis Based on a Registration of 3D Point Cloud, Journal of JSPS, Vol.81, No.12, pp.1113-1118 (2015).
(4) Yamaguchi, N., Tamori N. and Shiomi A. : A Lane Detection Method Using Adaptive Edge Preservative Smoothing, The transactions of the Institute of Electronics, Information and Communication Engineers. D-II Vol.J88-D-II, No.8, pp.1421-1431 (2005) (in Japanese).
(5) Muramatsu, M., Kawata T. and Kubo, F.: Traffic Lane Recognition by Image Segmentation and Object Recognition based on Rules, Technical report of IIEIE. PRMU Vol.98, No.334, pp.1-8 (1998) (in Japanese).
(6) Levinson, J., and Thrn., S. : Robust Vehicle Localization in Urban Environments Using Probabilistic Maps, IEEE International Conference on Robotics and Automation, pp.4372-4378 (2010).
(7) Badino, H., Huber, H., and Kanade, T. : Visual Topometric Localization, Proceedings of the IEEE Intelligent Vehicles Symposium, pp.794-799 (2011).
(8) Ziegler, J., Lategahn, H., Schreiber, M., Keller, C., G., Knöppel, C., Hipp, J., Haeusis, M., and Stiller, C. : Video Based Localization for BERTHA, Proceedings of IEEE Intelligent Vehicles Symposium, pp.1231-1238 (2014).
(9) Apostoloff, N., and Zelinsky, A. : Robust Vision based Lane Tracking using Multiple Cues and Particle Filtering, Proceedings of IEEE intelligent vehicles symposium, 7883374 (2003).
(10) Yamamoto, T., Kono, Y., and Ito, T : Vehicle motion estimation method using projective transformation of forward-images, Proceedings of JSAE Annual Congress, No.10-11, pp.7-12 (2011) (in Japanese).
(11) Haraishi, Y., Kazama, K., Hioki, T., Furusho, H. and Mouri, H. : Study on the edge detection algorithm using brightness gradient vector applied to the forward road image of the vehicle, Transactions of the Society of Automotive Engineers of Japan, Vol.45, No.5, pp.865-870 (2014) (in Japanese).
(12) Nishino, K. and Ikeuchi, K.: Robust simultaneous registration of multiple range images comparing a large number of points, Transactions of the Institute of Electronics, Information and Communication Engineers. D-II, Vol.J-85-D-II, No.9, pp.1413-1424 (2002) (in Japanese).

(13) Ikeuchi, K., Oishi, T., Takamatsu, J., Sagawa, R., Nakazawa, A., Kurazume, R., Nishino, K., Kamakura, M., and Okamoto, Y.: The Great Buddha Project: Digitally Achieving, Restoring, and Analyzing Culture Heritage Objects, International Journal of Computer Vision, Vol.75, No.1, pp.189-208 (2007).

(14) Higuchi, K., Hebert, M. and Ikeuchi, K.: Building 3-D models from unregistered range images, Proceedings of IEEE Conference on Robotics and Automation, pp.2248-2253 (1994).

(15) Ikeuchi, K.: Modeling from reality, Journal of the Robotics Society of Japan, Vol.16, No.6, pp.29-32 (1998).

(16) Kazama, K., Akagi, Y., Pongsathorn, R. and Mouri, H.: Fundamental study on road detection method using multilayered distance data with HOG and SVM, Journal of Robotics and Mechatronics, Vol.28, No.6, pp.870-877 (2016).

(17) Passani, M., Yebes J., and Bergasa, L.: Fast Pixelwise Road Inference Based on Uniformly Reweighted Belief Propagation, Proceedings of IEEE Intelligent Vehicles Symposium, pp.519-524 (2015).

(18) Lucas, B., D., and Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision, Proceedings of 7th Joint Conference on Artificial Intelligence (IJCAI), pp.121-130 (1981).

(19) Baker, S., and Matthews, I.: Lucas-Kanade 20 years On: A Unifying Framework, International Journal of Computer Vision, Vol.56, No. 5, pp.221-255 (2004).