Ego-localization using Streetscape Image Sequences from In-vehicle Cameras

Hiroyuki Uchiyama†, Daisuke Deguchi†, Tomokazu Takahashi‡, Ichiro Ide†, and Hiroshi Murase†

†Graduate School of Information Science, Nagoya University,
Furo-cho, Chikusa-ku, Nagoya-shi, Aichi 464–8601, Japan
‡Faculty of Economics and Information, Gifu Shotoku Gakuen University
Nakauzura 1–38, Gifu-shi, Gifu 500–8288, Japan
E-mail: †{uchiyama,ddeguchi,ide,murase}@murase.m.is.nagoya-u.ac.jp, ‡ttakahashi@gifu.shotoku.ac.jp

***Abstract*—This paper focuses on ego-localization using in-vehicle cameras. We propose a 2D ego-localization method using streetscape appearance as a feature of image matching and the triangulation of matching results. The image sequences of two in-vehicle cameras are matched to a database that contains a sequence of streetscape images and their corresponding positions. First, the proposed method searches for images similar to the input image from the database. Second, vehicle position is calculated based on triangulation using the positions stored in the database and the viewing directions of the two cameras. By assuming that the streetscape appearance changes continuously, a sequential image matching algorithm is used to improve the ego-localization accuracy. From experimental results, we confirmed that the proposed method surpasses the accuracy of a general GPS and achieved sufficient accuracy to be used for driving lane recognition.**

I. Introduction

Ego-localization is one of the most important functions for realizing accurate car navigation or such driver assistance schemes as a collision prevention system. Current car navigation systems use a general GPS to estimate vehicle positions. However, such a GPS has error of 5 to 30 meters due to reflections caused by buildings and occlusions of GPS signals from satellites. On the other hand, RTK-GPS can estimate vehicle positions more accurately, but it is sensitive to occlusion. To overcome these problems, several research groups have proposed ego-localization methods using such sensors as sonar [1], laser scanners [2], or cameras [3]. Since the camera resolution is high, some groups use in-vehicle cameras for ego-localization[4]. There are two approaches for this: landmark-based and appearance-based.

Landmark-based approach: This method consists of three steps: (1) store two- or three-dimensional landmark positions in a database, (2) detect landmarks from input images and compare them to the landmarks stored in the database, and (3) calculate self-position using the relationships between the landmark positions detected from the input images and the database. Both artificial features [5] and natural features (i.e., corners [6], and SIFT [7]) can be used in this approach. However, its performance heavily depends on the accuracy of the landmark detection process. Moreover, putting a huge number of artificial landmarks in a real environment is difficult.

Appearance-based approach: In this approach, the "appearance" of an object is used for comparing images [8], [9]. Here, an "appearance" is a view of an object from a certain position and direction. This approach consists of two steps: (1) storing images and corresponding positions in a database, and (2) finding an image having a similar appearance to the input image from the database and obtaining its corresponding position. J. Sato et al. [10] applied DP matching (Dynamic Programming Matching, DTW: Dynamic Time Warping) to match image sequences from omni-directional cameras for ego-localization. Using this idea for frame registration, H. Uchiyama et al. proposed an ego-localization method by matching an image sequence of a normal camera to that of an omni-directional camera [11]. Compared with the landmark-based approach, the appearance-based approach does not require geometrical object position. However, these methods cannot estimate a vehicle's lateral position (Fig. 1), since they assume that the trajectory of the self-positions is the same as the trajectory when the database was constructed.

To solve this problem, we propose a 2D ego-localization method using appearances as the features of image matching and the triangulation of the matched results. The proposed method consists of three steps:

1. Store a sequence of streetscape images and correspond-

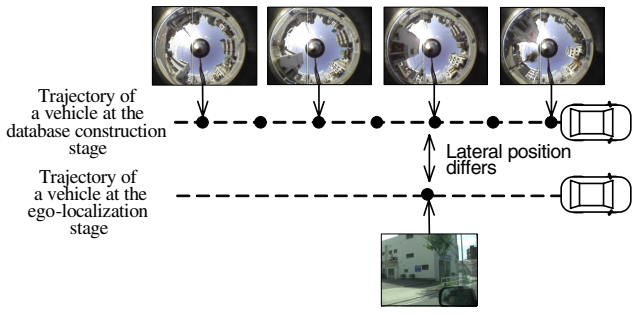


Fig. 1. Problem of appearance-based approach. Previous methods cannot estimate the lateral position of a vehicle, since they assume that the trajectory of self-positions is the same as the trajectory when database was constructed. We propose a 2D ego-localization method by triangulation using appearance as features of image matching.

978-1-4244-3504-3/09/$25.00 ©2009 IEEE

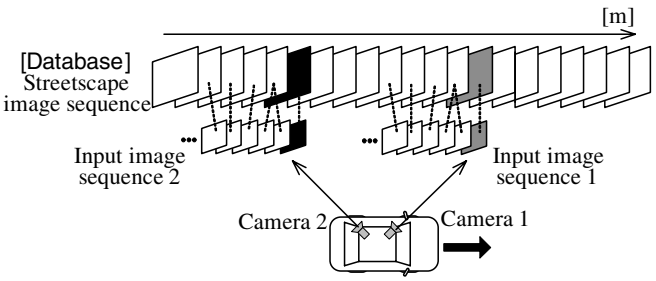


Fig. 2. Streetscape images can be obtained along the street. Therefore, a sequential image matching algorithm is used to improve accuracy.

ing positions in a database

2. Find images in the database that resemble the input images from the two cameras
   - A sequential image matching algorithm is used for improving the accuracy
3. Estimate the vehicle position by triangulation using the database positions and the viewing directions of the two cameras

In Section II, the details of the proposed method are described. Section III introduces the experiments, and Section IV discusses their results. Finally, we conclude the paper in Section V.

II. Ego-localization Using Streetscape Image Sequences

Our proposed method is composed of the following:

- Database construction stage
- Ego-localization stage

In the database construction stage, streetscape images and corresponding positions are obtained by a special vehicle equipped with an omni-directional camera. In the ego-localization stage, the proposed method compares input images with the image sequence of the database and finds positions corresponding to those input images. In this stage, we use two in-vehicle normal cameras with different viewing directions. Since the vehicle runs along a street, a sequential image matching algorithm is used to improve the precision of this process (Fig. 2). Finally, vehicle position is calculated by triangulation using the database positions and the viewing directions of the two cameras.

Figures 3(a), (b), and (c) show examples of streetscapes at positions P1, P2, and P3 obtained by an omni-directional camera. The rectangles indicate the views from normal cameras. As shown in the figures, we can assume that an image from camera 1 at P1 is similar to the streetscape image obtained at P2. Likewise, an image from camera 2 at P1 should resemble that at P3. From these assumptions, we calculate the coordinates of P1 based on triangulation using the viewing directions of cameras 1 and 2 and the coordinates of P2 and P3.

In this paper, the intrinsic parameters, the viewing directions, and the positions of the in-vehicle cameras are obtained before applying the above method. Figure 4 illustrates the flow

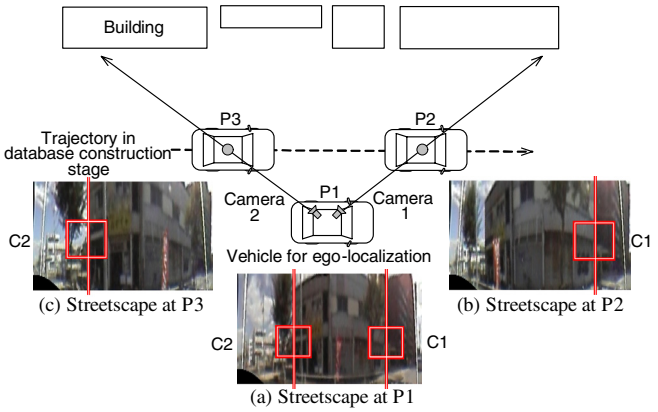


Fig. 3. Triangulation using streetscape images. Rectangles indicate views from normal cameras. Vehicle position P1 is estimated by database positions P2, P3, and camera directions.

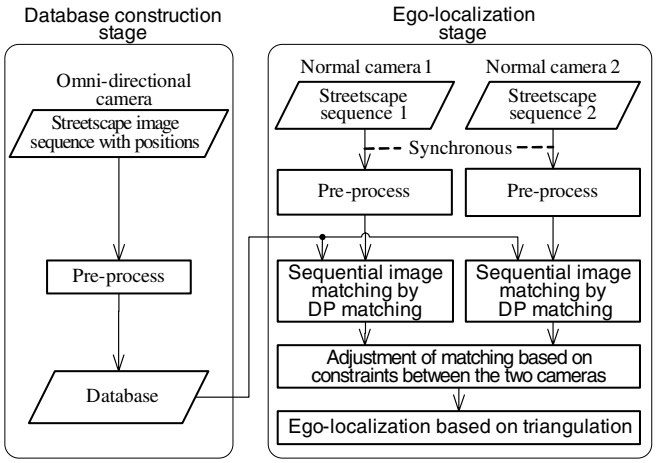


Fig. 4. Flow chart of proposed method. It consists of database construction and ego-localization stages.

chart of the proposed ego-localization method. The following sections describe the details of the method.

A. Database Construction

The database construction stage consists of the following three steps:

- Collection of streetscape images and corresponding positions by a special vehicle
- Regularization of frame velocity
- Spherical projection of streetscape images

The database used in our work consists of a sequence of images and their corresponding positions. These data are obtained by a special vehicle equipped with an omni-directional camera and a GPS. To simplify the matching process as much as possible, we assume that the velocity calculated by ego-localization roughly equals that in the database. To support this, the frame velocity of the sequence is regularized by resampling the image sequence so that the intervals between frames are constant in distance. Finally, the images are pro-

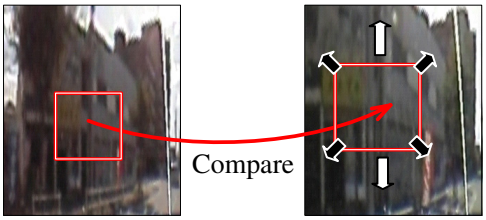


Fig. 5. Variation of appearance is approximated by translation and scaling. Rectangular in (a) shows input camera view, and (b) shows its corresponding area.

jected onto a spherical coordinate system to regularize the optical characteristics of the camera.

B. Ego-localization

As shown in Fig. 4, the ego-localization stage consists of the following three major steps:

1. Sequential image matching by DP matching
2. Adjustment of matching based on the constraints between the two cameras
3. Ego-localization based on triangulation

The proposed method uses two image sequences obtained from two in-vehicle cameras with different viewing directions. In the first step, each of these image sequences is compared with the image sequence in the database. This process is performed by DP matching. In the second step, the DP matching results are combined and adjusted to reduce the matching error of each camera. The positions stored in the database are obtained as adjustment results. In the third step, vehicle position is calculated using the positions stored in the database and the viewing directions of the two cameras.

1) Sequential Image Matching by DP matching

DP matching computes the cost between the current input frame and an arbitrary database frame. A modified edge-free DP matching is used for this DP matching. Before applying the following processes, the input images are projected onto a spherical coordinate system in the same way as in the database construction stage. All of the following processes are applied to the spherical polar coordinates.

a) Approximation of appearance variations caused by different camera positions: Variation of the appearances is observed in the input image when their view positions are different. Therefore, by assuming a weak-perspective camera, we approximate this variation as vertical translation on the spherical polar coordinates (Fig. 5). Similarly, image scaling is applied to approximate the variation caused by changing the distance between the camera and an object.

b) DP matching preserving the continuity of appearance variations: Both the input and database image sequences are matched with the translation and scaling parameters. Since a vehicle moves continuously, we assume that these parameters also change continuously. Under this assumption, matching of the two image sequences is calculated by DP matching between the one- and three-dimensional patterns used in [11]. This DP matching, which is an extension of matching for one- and two-dimensional patterns [12], provides us with the matching between input image frames $0, 1, \ldots, \tau_0$ and database image frames $0, 1, \ldots, t, \ldots, T$. As a result, the proposed method obtains the correspondence between input frame τ_0 and database frame t, together with its cost with the parameters of translation x and scaling s.

c) DP matching algorithm: For image comparison, the color components of the image are normalized to handle color variations caused by differences of devices and circumstances by equalizing the color histogram of the image.

The proposed method uses the Sum of Absolute Difference (SAD) of a color image to calculate distance $d(x, s, t, \tau)$ between input image $I_1(\tau)$ and database image $I_2(x, s, t)$, and the distance is defined as

$$d(x, s, t, \tau) = \mathrm{SAD}\{I_1(\tau), I_2(x, s, t)\}, \tag{1}$$

where x ($0 \leq x \leq X-1$) and s ($0 \leq s \leq S-1$) are translation and scaling parameters, respectively.

The details of the DP matching algorithm are shown below.
(Step 1) Initialize $g(\cdot)$ as

$$\begin{aligned} &g(-1, s, t, \tau) = g(x, -1, t, \tau) \\ &= g(X, s, t, \tau) = g(x, S, t, \tau) \\ &= \infty \end{aligned} \tag{2}$$

$$g(x, s, -i, \tau) = \infty \quad (i = 1, 2, \ldots, t_s). \tag{3}$$

(Step 2) Apply the following process for $\tau = 0, 1, \ldots, \tau_0$.

- $\tau = 0$

$$g(x, s, t, 0) = d(x, s, t, 0). \tag{4}$$

- $\tau \geq 1$

$$\begin{aligned} g(x, s, t, \tau) = \min_{\substack{-1 \leq x_p \leq 1 \\ -1 \leq s_p \leq 1 \\ 0 \leq t_p \leq t_s}} \{ & \\ & g(x - x_p, s - s_p, t - t_p, \tau - 1) \\ & + w(x_p, s_p, t_p, \tau) d(x, s, t, \tau)\}. \end{aligned} \tag{5}$$

(Step 3) Output cost $g(x, s, t, \tau_0)$ with parameters x and s.

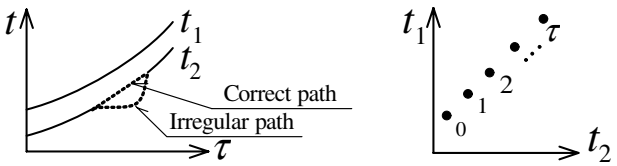
Here, $w(x_p, s_p, t_p, \tau)$ is a weight factor for penalizing the DP path. Based on the maximum velocity of the ego-localization vehicle and the intervals in the database, we adjust parameter t_s for controlling the slope of the DP path.

2) Adjustment of matching based on constraints between the two cameras

The results of the above edge-free DP matching may be inconsistent due to noise. To avoid this problem, our proposed method reduces the mismatch based on the following assumptions and the combination of the two DP matching results.

- Assumption 1: Vehicle's movement is continuous
- Assumption 2: Vehicle's forward movement is much larger than its lateral movement

We indicate the database frames of cameras 1 and 2 as t_1 and t_2, respectively. Assumption 1 induces that both t_1 and t_2 are continuous, while assumption 2 induces that the variation of the value $t_1 - t_2$ is kept small because the variation of



(a) (b)

Fig. 6. Adjustment of matching based on constraints between two cameras: (a) The proposed method tries to avoid irregular paths which do not satisfy the assumptions. (b) (t_1, t_2) for each τ should be aligned on the straight line.

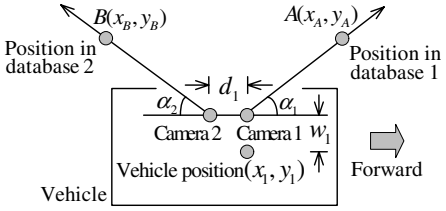


Fig. 7. Vehicle position (x_1, y_1) is calculated by database positions A and B, viewing directions α_1 and α_2, the distance between the vehicle's center line and the normal cameras w_1, and the distance between the two normal cameras d_1.

the distance between P2 and P3 in Fig. 3 is kept small. The proposed method tries to avoid irregular paths which do not satisfy the assumptions (Fig. 6 (a)).

The detail implementation is as follows. When a frame t in the database is matched with an input frame τ, its cost is defined as

$$f(t,\tau) = \min_{x,s} \frac{g(x,s,t,\tau)}{\tau+1}, \tag{6}$$

where $(\tau+1)$ is the regularization factor. This cost is calculated for both cameras 1 and 2, which are denoted as $f_1(t_1,\tau)$ and $f_2(t_2,\tau)$, respectively. The cost for selecting $\mathbf{t} = (t_1, t_2)$ is calculated by the combined distance

$$F(\mathbf{t},\tau) = f_1(t_1,\tau) + f_2(t_2,\tau). \tag{7}$$

Then, $\mathbf{t}_i$ are sought by minimizing the function

$$G(\tau) = \min_{\mathbf{t}_1,\mathbf{t}_2,\ldots,\mathbf{t}_\tau} \sum_{i=1}^{\tau} F(\mathbf{t}_i, i). \tag{8}$$

Here, Eq. (8) is minimized by changing $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_\tau$. From Assumptions 1 and 2, (t_1, t_2) for each τ should be aligned near the straight line (Fig. 6 (b)). Therefore, the combinations of $\mathbf{t}_i = (t_{1,i}, t_{2,i})$ should satisfy

$$\begin{aligned}(t_{1,i}, t_{2,i}) \in \Big\{ &(t_{1,i-1}+0 \quad , \quad t_{2,i-1}+0),\\ &(t_{1,i-1}+1 \quad , \quad t_{2,i-1}+1),\\ &(t_{1,i-1}+2 \quad , \quad t_{2,i-1}+2),\\ &(t_{1,i-1}+3 \quad , \quad t_{2,i-1}+3),\\ &(t_{1,i-1}+2 \quad , \quad t_{2,i-1}+3),\\ &(t_{1,i-1}+3 \quad , \quad t_{2,i-1}+2) \quad \Big\}.\end{aligned} \tag{9}$$

Equation (8) can be solved using dynamic programming [13]. Finally, we obtain the optimal frame pair $\mathbf{t}_\tau = (t_{1,\tau}, t_{2,\tau})$ for input frame τ.

3) Ego-localization Based on Triangulation

Vehicle position is estimated from the database positions obtained in the previous section as follows. Figure 7 shows database positions $\mathrm{A}(x_A, y_A)$ and $\mathrm{B}(x_B, y_B)$ that are obtained from the images of cameras 1 and 2. Exact vehicle position (x_1, y_1) is estimated by solving

$$y_1 + w_1 - y_A = (x_1 - x_A)\tan\alpha_1, \tag{10}$$

$$y_1 + w_1 - y_B = -(x_1 - x_B - d_1)\tan\alpha_2, \tag{11}$$

where α_1 and α_2 are the viewing directions of cameras 1 and 2, w_1 is the distance between the vehicle's center line and the cameras, and d_1 is the distance between the two normal cameras. Finally, the moving average of the lateral positions is calculated.

TABLE I
OMNI-DIRECTIONAL CAMERAS USED TO CREATE DATABASE

Omni-directional camera	Height h_2	Frame rate
(Omni I) VStone VS-C14N	1.5 m	30 fps
(Omni II) PointGreyResearch Ladybug2	2.0 m	15 fps

III. EXPERIMENTS

A. Conditions for Experiments

We conducted an experiment to evaluate the following:

- Estimation error on forward and lateral positions
- Recognition rate of driving lane

In addition, we compared the accuracy of the proposed method with a general GPS.

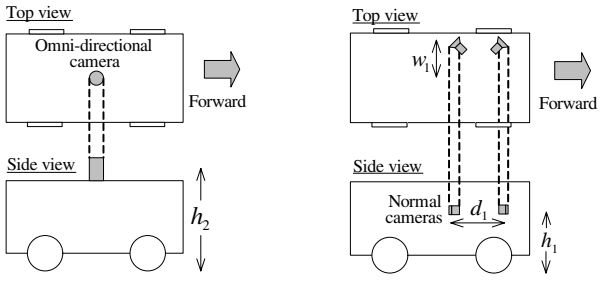
The positions of the in-vehicle cameras are shown in Fig. 8 and 9. The omni-directional camera used for the database construction was attached on top of the vehicle's roof. The two types of omni-directional cameras shown in TABLE I were tested to observe the effect of camera position and resolution difference.

Next, the conventional DV cameras used for the ego-localization stage were attached to the vehicle's side window. The focal lengths were 44.7 mm for the forward camera and 48 mm for the backward camera. The frame rates were 30 fps. The camera positions on the vehicle were $w_1 = 0.5$ m, $d_2 = 0.2$ m, and $h_1 = 1$ m.

TABLE II shows the datasets used in this experiment. During database acquisition, the vehicle kept running in the same lane. The length of the path was approximately 300 m. We prepared two types of datasets for evaluation. The image sequences for the database construction and the ego-localization stages were obtained under the following conditions.

- at the same time
- at different times

In the first condition, the cameras for both the database construction and ego-localization stages were mounted on the same vehicle at the same time to obtain identical trajectories in each stage. Datasets A-I and A-II, which were obtained in



(a) Camera for database construction was attached on top of the roof.

(b) Cameras for ego-localization were attached to side window.

Fig. 8. Position of in-vehicle cameras

Fig. 9. The vehicle equipped with Omni I and the normal cameras.

TABLE II
DATASETS PREPARED FOR EVALUATION

Dataset	Data acquisition for DB and ego-localization	Driving lane		Camera for DB
		for DB	for Ego-localization	
A-I	Same	Left	Left	Omni I
A-II	Same	Right	Right	Omni I
B-I	Different	Left	Left	Omni I
B-II	Different	Left	Right	Omni I
C-I	Different	Left	Left	Omni II
C-II	Different	Left	Right	Omni II

the first condition, were used to evaluate the estimation error of the lateral position within the driving lane. Datasets B-I, B-II, C-I, and C-II, obtained in the second condition, were used to evaluate the estimation error of the forward position and the recognition rates of the driving lanes.

For the ground truth and the database construction, we used the coordinates obtained from a general GPS, which were later corrected manually. The database images were resampled so that the vehicle position intervals become 0.4 m. The DP matching parameter was determined by a preliminary experiment; translation parameter x had 35 steps in 1.44°, and scaling parameter s had 13 steps in 0.03 times. Weighting factor $w(x_p, s_p, t_p, \tau)$ was 1.1 if $x_p \neq 0$ or $s_p \neq 0$, otherwise 1.0. We manually measured the viewing directions of the two in-vehicle cameras, and $\alpha_1 = 35°$ and $\alpha_2 = 32°$ were used for the ego-localization stage.

B. Results

Figure 10 shows the estimation error of the vehicle's forward positions. Figures 11 and 12 show the lateral estimation errors from datasets A-I and A-II. The positive values in Fig. 11 correspond to the vehicle's left side. Figure 13 shows examples of the estimated lateral positions from obtained datasets B-I and B-II. The recognition rates of the driving lane from B-I, B-II, C-I, and C-II are shown in TABLE III. They were calculated by approximating the width of the driving lane as 3.0 m. If lateral position x was estimated between -4.5 m to

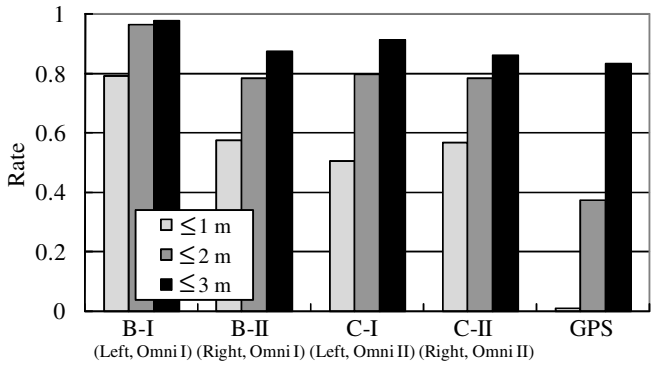
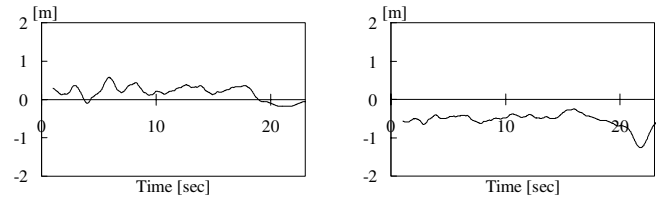


Fig. 10. Forward estimation error evaluated by datasets B-I, B-II, C-I, and C-II. Graph label shows dataset, driving lane in ego-localization stage, and camera for database construction stage.



(a) Left lane

(b) Right lane

Fig. 11. Lateral estimation error evaluated by datasets A-I and A-II. This distribution is shown in Fig. 12.

-1.5 m, it was classified as the right driving lane. Similarly, if $-1.5\,\text{m} \leq x \leq 1.5\,\text{m}$, it was classified as the same lane, and if $1.5\,\text{m} \leq x \leq 4.5\,\text{m}$, it was classified as the left lane. In Fig. 13, areas separated by dashed lines indicate lateral position x corresponding to each driving lane.

The computation time was 0.5 sec for processing one frame on Core 2 Quad Q9550 2.83 GHz CPU with 4 GB memory.

IV. DISCUSSION

This section discusses the effects caused by the differences of driving lanes and cameras.

A. Difference of Driving Lane

We observed that our proposed method's accuracy is much higher than that of a general GPS. For example, in Fig. 10, the forward location estimation by the proposed method with less than 2 m error was twice as much as that by a general GPS. On the other hand, the results of lateral location estimation from datasets A-I and A-II show that the maximum error is less than 1.5 m (Fig. 12). Figure 12 also shows that the estimation error of the right lane was larger than the left lane. Maybe the distance between the vehicle and the objects along the street affected the accuracy of the image matching.

From the results of datasets B-I, B-II, C-I, and C-II, we observed that accuracy decreases when the vehicle takes different lanes at the database construction and ego-localization stages (TABLE III, Fig. 10), perhaps because the appearance changed due to different viewing positions. The proposed method assumed a weak-perspective camera, but this assumption tends

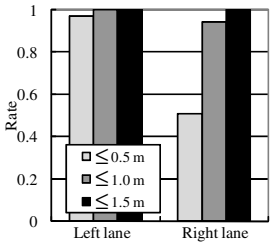


Fig. 12. Distribution of estimation error of lateral position evaluated by datasets A-I and A-II. Maximum error was less than 1.5 m.

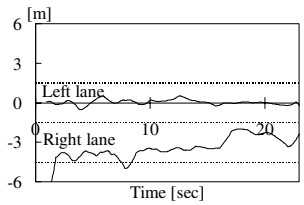


Fig. 13. Example of lateral vehicle position evaluated by datasets B-I and B-II. Dashed lines show boundary of driving lanes.

TABLE III
RECOGNITION RATE FOR DRIVING LANE (DATABASE: LEFT LANE)

Camera for DB	Left lane	Right lane
Omni I	100 % (dataset B-I)	93 % (dataset B-II)
Omni II	100 % (dataset C-I)	85 % (dataset C-II)

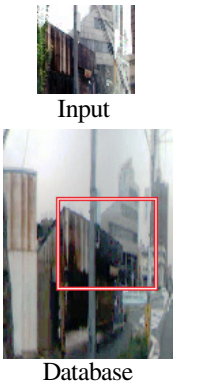


Fig. 14. Assumption of weak-perspective camera tends to fail when both nearby and distant objects are in an image.

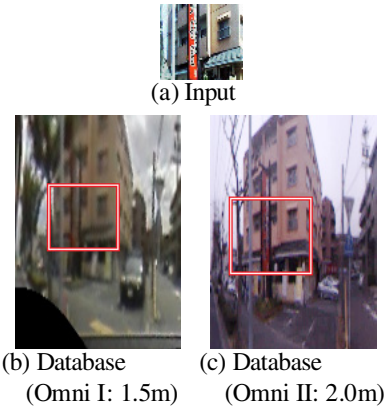


Fig. 15. Example of matched areas: camera positions in height and resolution were different between (b) and (c). Nevertheless accuracy was barely affected.

to fail when both nearby and distant objects appear in an image (Fig. 14). One solution for this problem is estimating vehicle position from the velocity and discarding the DP matching result when the system detects a decline of similarity between input and database images.

B. Differences between Cameras

Comparing the results of datasets B-I and B-II and datasets C-I and C-II, we observed that the influence of the height of the camera positions and the camera type is small (TABLE III, Fig. 10). The following two reasons might explain this.

First, although the appearances may differ, since the vehicle moves horizontally, the vertical edges played a more prominent role than the horizontal edges, especially in the forward location estimation (Fig. 15).

Second, as seen in Fig. 15, the resolution of (b) was lower than (c). Image matching accuracy did not decline significantly since the distance calculated by SAD is not affected much by resolution degradation.

V. CONCLUSION

We proposed a 2D ego-localization method using appearance as a feature of image matching and the triangulation of the matching result. To improve the ego-localization accuracy, a sequential image matching algorithm was used. As experimental results, maximum error of the lateral position less than 1.5 m was observed, and the recognition rate of the driving lane was about 85 to 100 %. Future work includes investigating a robust feature against changes of appearance and experiments using more data, also we will try to deal with a large streetscape image database.

ACKNOWLEDGMENTS

The authors would like to thank their colleagues for useful discussions. Parts of this research were supported by the Grant-In-Aid for Scientific Research from the Japan Society for Promotion of Science. This work was developed based on the MIST library (*http://mist.murase.m.is.nagoya-u.ac.jp*).

REFERENCES

[1] J. D. Tards, J. Neira, P. M. Newman, and J. J. Leonard, "Robust Mapping and Localization in Indoor Environments Using Sonar Data," Int. J. of Robotics Research, Vol. 21, Issue 4, pp. 311–330, April 2002.

[2] N. Shibuhisa, J. Sato, T. Takahashi, I. Ide, H. Murase, Y. Kojima, and A. Takahashi, "Accurate Vehicle Localization Using DTW between Range Data Map and Laser Scanner Data Sequences," Proc. 2007 IEEE Intelligent Vehicles Symposium (IV2007), pp. 975–980, July 2007.

[3] M. Agrawal and K. Konolige, "Real-time Localization in Outdoor Environments Using Stereo Vision and Inexpensive GPS," Proc. 18th Int. Conf. on Pattern Recognition (ICPR2006), Vol. 3, pp. 1063–1068, Aug. 2006.

[4] Y. Matsumoto, K. Sakai, M. Inaba, and H. Inoue, "View-Based Approach to Robot Navigation," Proc. Int. Conf. on Intelligent Robots and Systems (IROS2000), pp. 1702–1708, May 2000.

[5] M. Betke and L. Gurvits, "Mobile Robot Localization Using Landmarks," IEEE Trans. Robotics and Automation, Vol. 13, No. 2, pp. 251–263, April 1997.

[6] E. Royer, J. Bom, M. Dhome, B. Thuillot, M. Lhuillier, and F. Marmoiton, "Outdoor Autonomous Navigation Using Monocular Vision," Proc. Int. Conf. on Intelligent Robots and Systems (IROS2005), pp. 3395–3400, Aug. 2005.

[7] S. Se, D. Lowe, and J. Little, "Vision-based Global Localization and Mapping for Mobile Robots," IEEE Trans. Robotics, Vol. 21, No. 3, pp. 364–375, June 2005.

[8] H. Aihara, H. Iwasa, N. Yokoya, and H. Takemura, "Memory-based Self-localization Using Omnidirectional Images," Proc. 14th Int. Conf. Pattern Recognition (ICPR1998), Vol. 2, pp. 1799–1803, Aug. 1998.

[9] J. Y. Zheng and S. Tsuji, "Panoramic Representation for Route Recognition by a Mobile Robot," Int. J. Computer Vision, Vol. 9, No. 1, pp. 55–76, Oct. 1992.

[10] J. Sato, T. Takahashi, I. Ide, and H. Murase, "Change Detection in Streetscapes from GPS Coordinated Omni-Directional Image Sequences," Proc. 18th Int. Conf. on Pattern Recognition (ICPR2006), Vol. 4, pp. 935–938, Aug. 2006.

[11] H. Uchiyama, T. Takahashi, I. Ide, and H. Murase, "Frame Registration of In-Vehicle Normal Camera with Omni-Directional Camera for Self-Position Estimation," Proc. 3rd Int. Conf. on Innovative Computing, Information and Control (ICICIC2008), WS01–007, Jun. 2008.

[12] H. Sakoe, M. M. Ali, and Y. Katayama, "One Dimensional-Two Dimensional Dynamic Programming Matching Algorithm for Character Recognition," IEICE Trans. Information and Systems, Vol. E77-D, No. 9, pp. 1047–1054, Sep. 1994.

[13] R. Bellman, and S. Dreyfus, "Applied Dynamic Programming," Princeton University Press, 1962.