2014 IEEE Intelligent Vehicles Symposium (IV)
June 8-11, 2014. Dearborn, Michigan, USA

Single Camera Vehicle Localization using SURF Scale and Dynamic Time Warping

David Wong[1], Daisuke Deguchi[2], Ichiro Ide[1], Hiroshi Murase[1]

Abstract— **Vehicle ego-localization is an essential process for many driver assistance and autonomous driving systems. The traditional solution of GPS localization is often unreliable in urban environments where tall buildings can cause shadowing of the satellite signal and multipath propagation. Typical visual feature based localization methods rely on calculation of the fundamental matrix which can be unstable when the baseline is small.**

In this paper we propose a novel method which uses the scale of matched SURF image features and Dynamic Time Warping to perform stable localization. By comparing SURF feature scales between input images and a pre-constructed database, stable localization is achieved without the need to calculate the fundamental matrix. In addition, 3D information is added to the database feature points in order to perform lateral localization, and therefore lane recognition.

From experimental data captured from real traffic environments, we show how the proposed system can provide high localization accuracy relative to an image database, and can also perform lateral localization to recognize the vehicle's current lane.

I. Introduction

Ego-localization is a critical stage in not just in-car navigation systems, but also a required component for many of the emerging driver assistance and obstacle avoidance methods. As we move towards fully autonomous driving ability, efficient and effective ego-localization is of increasing importance, as current GPS systems rarely manage an accuracy of 5 metres even in ideal environments, and often much lower in urban situations. Even expensive, high precision RTK-GPS combined with IMU sensors can be sensitive to the occlusions common in city driving environments. For tasks such as lane recognition and obstacle avoidance, higher precision in all environments is required.

There are many proposed ego-localization methods which use a variety of different sensors, including SONAR [1], LASER scanners [2], Inertial Measurement Units (IMU), cameras [3] or various combinations of the above [4], [5]. There is an increasing number of systems that use a pre-constructed database which is then localized against [5], [6], [7], [8]. Most current systems either use several cameras or a variety of sensors for localization, or are unable to perform lateral positioning so can not recognize which lane is currently being used by the vehicle.

In this paper we propose a method for ego-localization using a pre-constructed database and an in-vehicle monocular camera, with no other supporting sensors. In order to achieve this, we present two main novel contributions:

1) Monitoring the scale difference between SURF [9] features in the input and database images within a Dynamic Time Warping (DTW) method to achieve stable localization in the direction of motion, without calculation of the fundamental or essential matrices
2) Embedding of 3D information into the feature points of the database, so that changes in lateral position can be triangulated, and therefore lane recognition performed

We show that ego-localization and lane recognition can be achieved using only a single in-vehicle camera and a pre-constructed database. We demonstrate the performance of our system in a typical urban traffic environment, and show that using feature scale changes is a robust way to find the closest database image and therefore localize the current image. The use of a database with 3D feature point information allows us to reliably perform lane recognition.

This paper is organized as follows: In Section II we give a brief overview of related research. We describe the proposed method in more detail in Section III and experimental results are presented in Section IV. We discuss the results in Section V before concluding in Section VI.

II. Related Work

The ego-localization problem is one shared by automotive applications [5], [6], [8], and robotics [1], [10] where it is more often posed as the Simultaneous Localization and Mapping (SLAM) problem [11]. SLAM has been a very active research area in the robotics community where unknown environments must be mapped as the robot is localized within the dynamically updated map. There have been several successful adaptations using monocular vision [10], [12], [13], which employ visual odometry by way of structure from motion to monitor camera movement and current localization. These methods typically use extended Kalmann filter based [12] or, more commonly, pose graph approaches [10], [13]. SLAM localization methods do not easily scale to the large environments that are typical of automotive applications, where the vehicle position within a known map is of interest rather than self-localization relative to a map of previously visited areas.

For automotive ego-localization, similar monocular methods have been employed which separate the mapping step from localization by using a pre-constructed database [5], [8] or image databases such as Google Street View [14]. These methods perform complete localization relative to the database images, which can enable high accuracy—for

[1]David Wong, Ichiro Ide, and Hiroshi Murase are with the Graduate School of Information Science, Nagoya University, Japan
{davidw, ide, murase}@murase.m.is.nagoya-u.ac.jp
[2]Daisuke Deguchi is with the Information and Communications Headquarters, Nagoya University, Japan ddeguchi@nagoya-u.jp

example, up to 10 cm precision when combined with an IMU [5]. They also usually employ supporting sensors to achieve localization—either an IMU [5] or odometry information [14]. A simpler approach is to localize against the closest database image, of known location, using DTW (or Dynamic Programming) to remove temporal differences between input and database image streams [6], [15]. The matching between sample images and those in the database can be performed by using a kind of template matching [15], or by using a low bit-rate image sequence instead of single images [16], which is stable in varying environments but does not provide high localization accuracy. Novel feature based methods have also been proposed [7], where the epipole calculated from matched features between images is tracked. The method introduced in reference [7] uses the position of the epipole as a cost measure for comparing image positions. The epipole moves away from the vanishing point as the image positions become similar. While effective, this technique requires the calculation of the fundamental matrix so can be unstable when the baseline between the query and database images is small. The above Dynamic Time Warping based methods do not calculate the lane position relative to the database images. This has been attempted to be solved by using two cameras and triangulation [6]. Image matching is also performed by some SLAM methods for loop closure, demonstrated effectively using Bag of Features [13]. While these techniques are effective at matching similar images, they are not capable of spatially arranging them in order without complete frame-by-frame visual pose estimation.

III. Proposed Localization Method

This paper describes a method of ego-localization using a pre-processed image database. Images captured by an in-vehicle camera in the localization step are compared to images from the database using DTW to compensate for speed differences in the two image streams. If two images have the same viewing direction, their corresponding SURF feature points will have a similar scale when the capture positions were spatially close. As distance between the images increases, the difference in corresponding feature scales also increase. The proposed method makes use of this change to match images between the input image and the database, therefore localizing the input image in the direction of motion. Only the scale change between the matched features is used, so there is no need to calculate fundamental matrices. This allows matching even when the baseline between input and database images is small.

Lane recognition is performed by using the change in pixel co-ordinates of the features in the input image when compared to the corresponding features of the matched database image. The pixel co-ordinate change and known 3D points of the features from the database image are used to estimate the lateral translation between the two images and therefore recognize the vehicle's current lane.

The process is described in more detail below. Section A describes the database construction step, and Sections B and C detail the implementation of the database relative localization and lane recognition, respectively. An overview of the proposed system is presented in Fig. 1.

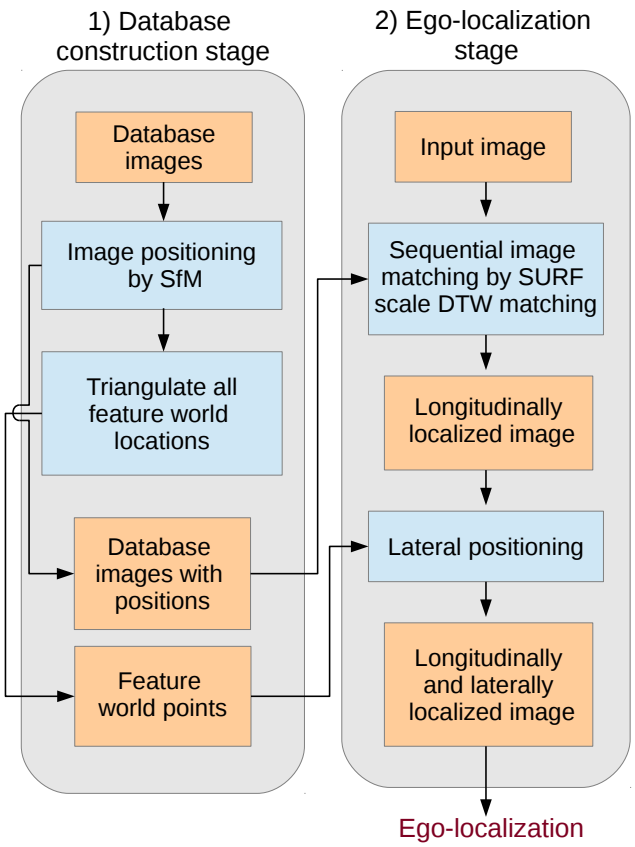

Fig. 1. A flow chart outlining the proposed method. The two main stages are 1) database construction, and 2) ego-localization by DTW matching the images from the vehicle to be localized to the database image stream.

A. Database Construction

The database used by this method consists of series of images with corresponding world locations for each image. Each image has a set of extracted SURF features, and 3D world positions for each feature. The database construction step is performed only once, and the resulting database is then used as a map for localization. GPS localization data for each image is included in the database.

The database construction step follows a method fairly similar to standard visual odometry and visual SLAM techniques [5], [10]. SURF features are extracted from the captured images. Feature matching is performed between spatially close images with each new image being matched to the nearest six images. The feature matches are pruned using RANSAC and geometric constraints. The 3D locations of all of the feature points are calculated using a Pose N-Point algorithm based on Levenberg-Marquardt optimization [17]. A sparse bundle adjuster [18] refines matches and also determines the camera coefficients, calculating 3D feature points to minimize re-projection error. As a new database image is entered, it is used to create a new bundle with the

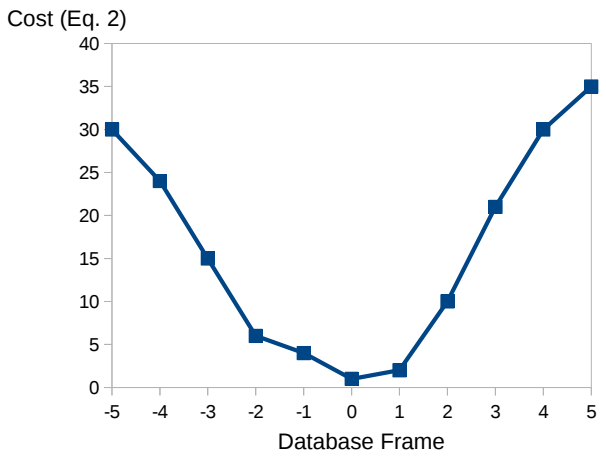

Fig. 2. The cost measure used for DTW matching (summed scale) for a sample image. The correct image match is database frame 0 in this graph.

next closest images, and the 3D locations of its feature points are calculated.

The completed database is a series of images with corresponding camera positions and SURF keypoints with descriptors. Each SURF feature which has a corresponding calculated 3D location is referenced to a point in a database of 3D world points; naturally each world point has been constructed from many different database images.

B. Localization

The localization requires only the pre-constructed database and input images from an in-vehicle camera. The localization process can be broken into two main steps:

1) SURF feature extraction and matching
2) Sequential image matching to the database images using summed scale of SURF points and DTW matching

The result is a 2D position for the camera for the current image, corresponding to the closest database image location.

1) SURF feature extraction and matching: Extracted SURF features are matched to features from the series of the street map images, starting from the last matched image in the sequence. The proposed method relies quite heavily on a reasonable number of correct matches. To keep the localization step simple and efficient, RANSAC pruning of outlier feature matching is avoided. Instead, we propose a simple weighted matching cost which matches features based on known constraints. The views in both streams are forward looking and the camera heights are considered constant, so good image match candidates will have similar y pixel coordinates and a limited change in scale and feature response. Based on these properties, a weighted criteria is used for determining likely inlier matches. A spatial constraint is applied so that each potential match candidate is only searched for in a region with pixel values close to where the feature was located in the query image, particularly in the y direction of the image plane. The candidate features must also be from the same octave. Then the best match for the query image feature f_τ is calculated by finding the database image feature f_i within the set of N features $i = 1, 2, ..., N$ which minimizes the following equation:

$$m(i) = w_s|s(f_\tau) - s(f_i)| + w_r|r(f_\tau) - r(f_i)| + w_d(\mathrm{SSD}(f_\tau, f_i)), \quad (1)$$

where $s(f)$ is the feature scale, $r(f)$ is the feature response, and $\mathrm{SSD}(f_1, f_2)$ is the standard sum of squared differences of the feature descriptors. The weights w_s, w_d, w_r are adjusted to give a strong inlier set while maintaining a high number of matched features.

2) DTW matching: DTW matching computes the cost between the current input image and an arbitrary street map image. We propose a cost measure based on the summed scale of matched SURF features. Matched SURF features that are extracted at the same octave [9] may vary in scale if there is a translation between the two cameras; we make use of the scale differences to match images which are closest to each other. For a set of database images $I_1 = \{t \mid 0 \leq t \leq T_1\}$ and input images $I_2 = \{\tau \mid 0 \leq \tau \leq T_2\}$ we take the latest input image $I_2(\tau)$ for localization. A subset of the database images, $\tilde{I} \subset I_1 \rightarrow \tilde{t} \in \tilde{I}$ is selected for cost minimization. This is done by calculating the input image feature matches with sequential database images, starting with the previously matched database image and continuing until the number of matched features falls below a threshold. Within the resulting subset of database images, only the individual feature matches that are consistent throughout the whole subset are used. This results in a set of $N_{\tilde{t},\tau}$ matched features f in each subset database image $\tilde{I}(\tilde{t})$ and the input image $I_2(\tau)$. The cost of each image match $g(\tilde{t}, \tau)$, is calculated by summing the absolute feature scale differences as follows:

$$g(\tilde{t}, \tau) = \sum_{i=0}^{N_{\tilde{t},\tau}} |s(f_{\tilde{t},i}) - s(f_{\tau,i})|, \quad (2)$$

where $s(f)$ is the scale parameter extracted from the relevant SURF feature. The database image which minimizes $g(\tilde{t}, \tau)$ is deemed to be the closest location to the input image, providing localization in the direction of motion. Fig. 2 shows the cost of a sample image against a series of database images.

C. Lane Recognition

The ego-localization method described above provides localization in the direction of motion. However, variation in the lateral positioning of the vehicle is also of interest, particularly when considering navigation applications where the current lane may be of importance. We propose lane recognition and approximate lateral localization using the same SURF features that were used previously without requiring a complete triangulation or visual pose estimation. Fig. 3 illustrates the lane recognition problem.

For lane position tracking, we assume that the database relative localization has been successful and accurate. We return to using all of the feature matches between the current input image and the database image that we previously

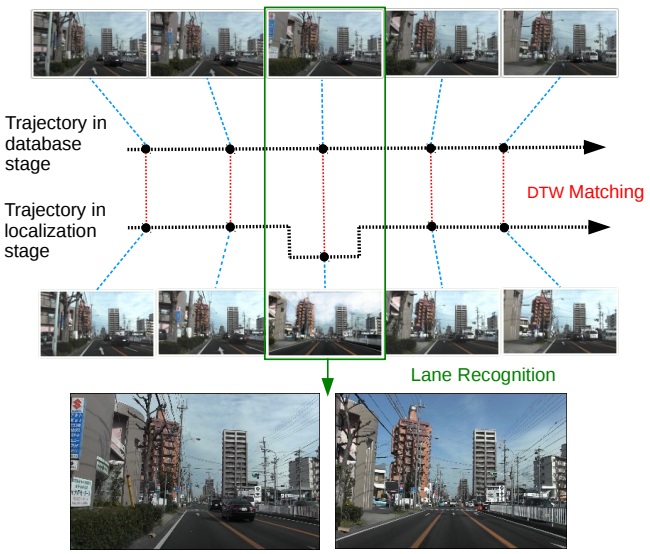

Fig. 3. Illustration of the localization process when the database and input image stream lanes are different.

selected by DTW matching. The following assumptions are made for typical traffic situations:

- Variation in orientation between the two views is purely in the x direction of both image planes
- Both images were captured with the camera facing in the direction of primary motion
- The height of the camera is constant between image streams

We have previously calculated the 3D world positions of feature points in the database. An example world point $\mathbf{x}_w$ will be projected to pixel points $\mathbf{x}_{\mathrm{db}}$ and $\mathbf{x}_\tau$ for the database and input image planes respectively. More generally, if we include the camera matrix $\mathbf{K}$ and the relative motion between the two images described by the rotation matrix $\mathbf{R}$ and translation vector $\mathbf{T}$,

$$\begin{bmatrix} x_\tau \\ y_\tau \\ 1 \end{bmatrix} = \mathbf{K}[\mathbf{R}|\mathbf{T}] \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \tag{3}$$

where we consider here, for convenience of notation, that the world point has been transferred into the co-ordinates of the database image. The camera intrinsics matrix $\mathbf{K}$ has been previously calculated in the database construction step. Asserting the above assumptions, the motion can be separated into a simple translation in the x direction, Δx. This allows Eq. (3) to be arranged to calculate Δx using the 3D world points from the database, (x_w, y_w, z_w), and the pixel x location of the matched feature, x_τ, as follows:

$$\Delta x = \frac{1}{f_x}\left(z_w x_\tau - f_x x_w - s y_w + u_0 z_w\right) \tag{4}$$

where f_x, s, u_0, are parameters from the calibration matrix. The value of Δx is averaged over all of the matched features.

TABLE I
IMAGE SEQUENCES

Sequence	Road lanes used
DB (database)	Left only
A	Left only
B	Right only
C	Both (3 lane changes)

When the averaged value reaches a threshold, it signifies a lateral motion approximately equal to the lane width allowing recognition of the lane through which the vehicle is currently travelling.

IV. EXPERIMENTS

Testing of the proposed method was carried out in an urban environment which included a variety of buildings, traffic, lighting variations and lane changes.

The database and ego-localization streams were captured using a Canon iVIS HF G20 video camera attached to the inside windscreen of the vehicle, facing forwards. Images were captured at a rate of 23.97 frames per second, at varying speeds up to 50 km/h. Images were captured in color but later converted to grayscale. Several passes were made of the same stretch of road (over about 1 km). Table I gives a summary of the lane positions of the road passes. All passes were completed within the same day, with varying lighting and traffic conditions.

The database construction gives a theoretical potential localization accuracy of about 60 cm, when captured at 50 km/h and 23.97 frames per second. GPS capable of this level of accuracy was not available for the database construction, so it is difficult to quantify the exact precision achieved by localizing each image from the structure from motion and bundle adjustment process. However, with the correct sensor suite, it should be quite possible to construct a database with a high accuracy for image localizations. Therefore, we focus on the performance of the ego-localization relative to the database images.

For the following results, sample images were taken from each image sequence and visually compared to check for correct matching. For each sequence, 300 images were sampled. The samples were chosen by randomly selecting ten sets of 30 successive images. This sampling method was used so that lane changes and partial lane changes were sampled and represented in the results.

A. Ego-localization in the Direction of Motion

The sequential image matching by DTW matching of SURF scale changes was evaluated on all input image sequences. The results are shown in Table II.

In Sequence A, when the vehicle was travelling in the same lane as the lane used for database construction, localization error was consistently lower than two frames (matched to an image within two frames of the correct database image). Image matching was exact, corresponding

to an accuracy of 60 cm, 94% of the time. For Sequence B, where the vehicle was in a different lane to the one used in database construction, the exact frame matching rate fell to 82% but was always within four frames. It was noted that where incorrect image matching occurred, the system quickly recovered to higher frame recognition rates, with incorrect matches always corrected within four frames. While the constant lane image sequences showed high correct match rates and therefore high localization accuracy, in Sequence C where lane changes occurred, there were some mismatches as the vehicle changed lanes. Even where an exact match was not found, the result was always within six frames. It appears that the mismatches that occurred during lane changes were caused by the change in viewing direction as the vehicle moved into a different lane, so were particularly apparent when the vehicle first started the lane change. As the new lane was entered, the images were once again correctly matched.

Localization relative to the database was also performed using the inverse of the number of matched features for the DTW matching cost, as a comparative method. The cost measure in Eq. 2 was replaced with the following:

$$g(\tilde{t},\tau) = 1/N_{\tilde{t},\tau}. \tag{5}$$

The image matching accuracy of the comparative method is compared to the results of the proposed method in Fig. 4. The use of feature scale is very effective for image matching and localization when the view direction is constant. Using the scale of matched features gives a more robust distance measure between the input and database images, resulting in better image matching performance. Sample image matching for both the SURF scale method and number of matched features method is shown in Fig. 5.

B. Lane Recognition Results

The lane recognition performance of the algorithm was tested on all three image sequences. The results are presented in Table III, which shows the percentage of correct lane recognitions for each sequence. For Sequence A, no lane changes relative to the database took place (both used only the left-hand lane). The lateral lane localization system also correctly determined that the lane was the same 100% of the time. The lateral position was monitored and no significant motion relative to the database was detected. For Sequence B, the lateral positioning algorithm correctly detected that the traversed lane was the right-hand lane throughout. There were 15 images out of the sampled 300 where a lane change was incorrectly detected as a result of outlier features being used in the lateral positioning calculation. With monitoring of previous image lane positioning, the erroneous lane recognitions could be easily filtered, as the incorrect recognitions only affected isolated images representing impossible lane changes. In Sequence C, when the vehicle was between lanes, the higher error in image localization as mentioned in Section A resulted in a higher lateral localization error. Lane recognition was successfully performed when the lane recognition results stabilized as the vehicle completely entered the lane. There is ambiguity as to which lane the vehicle is in when it is between lanes, so the presented lane recognition results only include image samples from when the vehicle was completely in a particular lane.

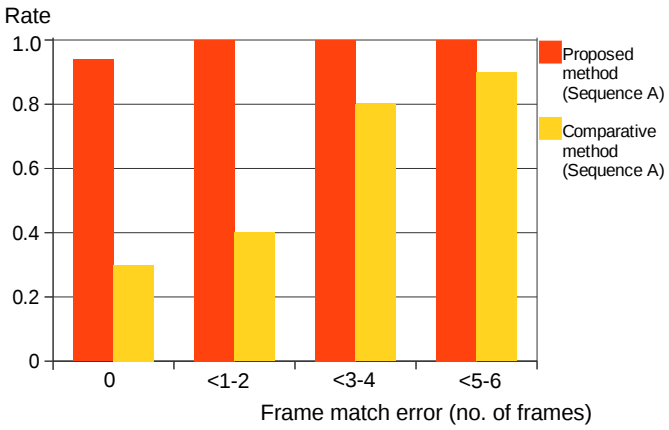

Fig. 4. Comparison of the image match accuracy rates for the proposed method and the comparison method, which uses the number of matched features for image matching. Sequence A was used for this comparison.

Fig. 5. Sample images showing DTW matching of images. The central column is the database, and the one on left the corresponding matched images using the proposed method. The column on the right shows the matched images using the comparative method. The numbers in the top right of the images are the sequence frame numbers, showing how DTW matching absorbs differences in vehicle speeds between the database and localization image sequences.

V. Discussion

In this section we discuss the performance of the proposed method and identify areas where improvements could be made.

The lack of accurate ground truth data made it difficult to quantitatively analyse the accuracy of the system. There are a number of factors that affect localization accuracy.

TABLE II
IMAGE MATCH RESULTS

	Image match error (no. of frames)			
	0 (exact)	<1–2	<3–4	<5–6
Sequence A	94%	100%	100%	100%
Sequence B	82%	94%	100%	100%
Sequence C	80%	88%	97%	100%

TABLE III
LANE RECOGNITION RESULTS

	Correct lane recognition rate
Sequence A	100%
Sequence B	95%
Sequence C	84%

The camera frame rate and vehicle speed of the database sequence determine the absolute accuracy of localization in the direction of motion. Since the proposed system finds the closest database match to an input query image, the finer the spacing of the database images (and the more accurately their positions are known), the higher the accuracy of localization. Future research will include the construction of datasets with image capture positions known to a high degree of accuracy, using a suite of sensors such as GPS and odometry. This would allow better analysis of the metric accuracy of the system. We presented results in terms of image matching accuracy, which was validated by visual checking of matches; performance in terms of metric error when compared to a ground truth would be preferable.

For lane recognition, the accuracy is highly dependant on the precision of the localization in the direction of motion. The accuracy considerations for localization in the direction of motion are also relevant for lateral positioning. The average lane width of roads in the sample image sequences were approximately 3.0–3.5 metres, so lane recognition requires lateral positioning at a resolution not substantially different from the typical accuracy of the image matching localization on which it depends. Therefore the stability of the lane recognition step could be greatly enhanced by a database captured at a higher frame rate or slower speed. The image sequences used for the experiments all included the use of only two lanes. With a wider angle camera lens the system could also be tested on three or more lanes.

VI. CONCLUSION

We proposed a method for ego-localization using summed SURF scale changes across matched features as a cost measure for sequential image matching against a pre-constructed database. Lateral positioning and therefore lane recognition was achieved by including 3D feature point information in the database. The experimental results show that a database image matching accuracy within two consecutive images can be achieved at a rate of 88% or higher. Lane recognition rates vary between 84 to 100%. Future work includes improved database construction and performance analysis using image sequences with high accuracy GPS and odometry information.

ACKNOWLEDGMENTS

Parts of this research were supported by JST CREST, JST COI, JSPS Grant-in-Aid for Scientific Research, and the Japanese Ministry of Education, Culture, Sports, Science, and Technology (MEXT).

REFERENCES

[1] B.-S. Choi and J.-J. Lee, "Mobile robot localization in indoor environment using RFID and sonar fusion system," in *Proc. 2009 IEEE International Conference on Intelligent Robots and Systems (IROS2009)*, 2009, pp. 2039–2044.

[2] L. Armesto and J. Tornero, "Robust and efficient mobile robot self-localization using laser scanner and geometrical maps," in *Proc. 2006 IEEE International Conference on Intelligent Robots and Systems (IROS2006)*, 2006, pp. 3080–3085.

[3] P. Jeong and S. Nedevschi, "Reliable localization and map building based on visual odometry and ego motion model in dynamic environment," vol. 1, 2010, pp. 316–321.

[4] A. Bacha, A. Redouane, D. Gruyer, and A. Lambert, "A robust hybrid multisource data fusion approach for vehicle localization," *Positioning*, vol. 4, no. 4, pp. 271–281, 2013.

[5] H. Lategahn, M. Schreiber, J. Ziegler, and C. Stiller, "Urban localization with camera and inertial measurement unit," in *Proc. 2013 IEEE Intelligent Vehicles Symposium (IV2013)*, 2013, pp. 719–724.

[6] H. Uchiyama, D. Deguchi, T. Takahashi, I. Ide, and H. Murase, "Ego-localization using streetscape image sequences from in-vehicle cameras," in *Proc. 2009 IEEE Intelligent Vehicles Symposium (IV2009)*, 2009, pp. 185–190.

[7] H. Kyutoku, T. Takahashi, Y. Mekada, I. Ide, and H. Murase, "On-road obstacle detection by comparing present and past in-vehicle camera images," in *Proc. 12th IAPR Conference on Machine Vision Applications (MVA2011)*, 2011, pp. 357–360.

[8] S. Nedevschi, V. Popescu, R. Danescu, T. Marita, and F. Oniga, "Accurate ego-vehicle global localization at intersections through alignment of visual data with digital map," *IEEE Trans. Intelligent Transportation Systems*, vol. 14, no. 2, pp. 673–687, 2013.

[9] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[10] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Monocular vision based SLAM for mobile robots," in *Proc. 18th International Conference of Pattern Recognition (ICPR2006)*, vol. 3, 2006, pp. 1027–1031.

[11] H. F. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[12] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[13] T. Botterill, S. Mills, and R. D. Green, "Bag-of-words-driven, single-camera simultaneous localization and mapping," *Journal of Field Robotics*, vol. 28, no. 2, pp. 204–226, 2011.

[14] H. Badino, D. F. Huber, and T. Kanade, "Real-time topometric localization," in *Proc. 2012 IEEE International Conference on Robotics and Automation (ICRA2012)*, 2012, pp. 1635–1642.

[15] J. Sato, T. Takahashi, I. Ide, and H. Murase, "Change detection in streetscapes from GPS coordinated omni-directional image sequences," in *Proc. 18th International Conference of Pattern Recognition (ICPR2006)*, 2006, pp. 935–938.

[16] M. Milford, "Visual route recognition with a handful of bits," in *Proc. 2012 Robotics: Science and Systems VIII*, 2012, pp. 297–304.

[17] O. Enqvist and F. Kahl, "Robust optimal pose estimation," in *Proc. 10th European Conference on Computer Vision (ECCV2008)*, 2008, pp. 141–153.

[18] M. I. A. Lourakis and A. A. Argyros, "SBA: A software package for generic sparse bundle adjustment," *ACM Trans. Mathematical Software*, vol. 36, no. 1, pp. 1–30, 2009.