

A Dependence Maximization Approach towards Street Map-based Localization

Kiyoshi Irie^{1,2}

Masashi Sugiyama³

Masahiro Tomono¹

Abstract—In this paper, we present a novel approach to 2D street map-based localization for mobile robots that navigate mainly in urban sidewalk environments. Recently, localization based on the map built by Simultaneous Localization and Mapping (SLAM) has been widely used with great success. However, such methods limit robot navigation to environments whose maps are prebuilt. In other words, robots cannot navigate in environments that they have not previously visited. We aim to relax the restriction by employing existing 2D street maps for localization. Finding an exact match between sensor data and a street map is challenging because, unlike maps built by robots, street maps lack detailed information about the environment (such as height and color). Our approach to coping with this difficulty is to maximize statistical dependence between sensor data and the map, and localization is achieved through maximization of a Mutual Information-based criterion. Our method employs a computationally efficient estimator of Squared-loss Mutual Information through which we achieved near real-time performance. The effectiveness of our method is evaluated through localization experiments using real-world data sets.

I. INTRODUCTION

We aim to realize autonomous navigation of mobile robots in urban sidewalk environments. Over the past few decades, mobile robot navigation has been studied intensively, and several authors have reported successful autonomous navigation in large-scale real environments [1] [2]. However, their localization methods use precise maps of the environment built by Simultaneous Localization and Mapping (SLAM) techniques, in which sensor data are typically collected by manually operated robots. The cost involved in the construction and maintenance of such maps will increase rapidly as the size of the environment increases. This research aims to develop a localization system for a mobile robot by using existing street maps. The utilization of existing street maps eliminates the additional cost required for building maps that are dedicated to robots, and enables them to localize themselves in unfamiliar environments.

One of the significant challenges in using 2D street maps for localization is matching sensor data with a map. This complexity did not influence traditional robot localization because, the robot localizes itself on a map built from its

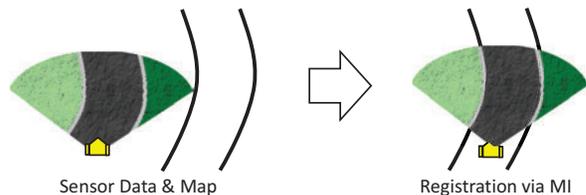


Fig. 1. Concept diagram of our localization method.

onboard sensor data [3] [4] [5]. Such maps are easy to be matched with sensor data collected during navigation.

However, typical 2D street maps do not provide detailed information about the environment. For example, we cannot predict colors of the road or the height of the curbs between roadways and sidewalks or even the existence of curbs from street maps. This makes correlating sensor data with maps difficult.

Majority of existing street map-based localization methods are designed for assisting human car drivers (e.g. car navigation systems) [6] [7] [8] [9]. Such methods are not influenced by the abovementioned difficulty because, localization is performed by matching a vehicle trajectory with road network information extracted from street maps, assuming that the vehicle is always on the road network. However, the precise position with respect to the road, which is crucial information for autonomous navigation, cannot be estimated.

Although several authors have already proposed using street maps for localization aiming to autonomous navigation, their methods are somewhat limited because of not addressing the matching issue. For example, Chausse *et al.* proposed to detect lane markings from images and match them with the map [10]. Morales *et al.* proposed to detect road center by laser measurements and match them with the map [11]. Those methods are not directly applicable to varying sidewalk environments where robust detection of road boundaries is not straightforward. Hentschel and Wagner proposed to detect building boundaries and match them with the map [12]. The advantage of using building boundaries is that they can be easily detected by a laser scanner. However, the method is only applicable in environments where buildings are sufficiently observable.

On the other hand, the matching issue was discussed by Irie and Tomono [13], in their paper they proposed a method that combines prior information and object recognition to match sensor data with a street map. The method exploits prior knowledge about objects existing in the environment. The knowledge is expressed as prior probabilities bundled with semantic labels in the map such as *roadway* and

¹Kiyoshi Irie and Masahiro Tomono are with Future Robotics Technology Center, Chiba Institute of Technology, 2-17-1 Tsudanuma, Narashino-shi, Chiba 275-0016, Japan {irie, tomono}@furo.org

²Department of Computer Science, Tokyo Institute of Technology, 2-12-1 O-okayama, Meguro-ku, Tokyo 152-8552, Japan

³Masashi Sugiyama is with Department of Complexity Science and Engineering, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan sugi@k.u-tokyo.ac.jp

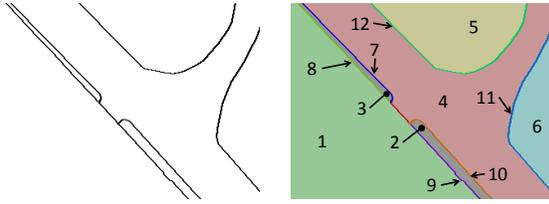


Fig. 2. Left: an example of maps used in proposed method. Right: internal representation of the map. Different colors indicate different IDs.

building. The limitation of their method is, however, that it is not applicable to unfamiliar environments where we have no prior information about the kind of existing objects or their appearance.

In this paper, we propose a novel localization framework that relaxes the limitations in the previous methods. Our method does not explicitly detect road boundaries nor use prior information. Instead, sensor data is directly matched to a map using a variant of Mutual Information (MI). The concept of our method, which is inspired by image registration using MI, is depicted in Fig. 1. Although MI has already been used to register multimodal sensor data [14] [15] [16], to our knowledge, it has never been applied to street map-based localization. Since the calculation of the original MI is computationally expensive, we employ a squared-loss version of MI for efficient computation.

The differences between our method and existing methods are summarized as follows:

- We do not assume that the robot is always on the road.
- Neither road boundaries nor building boundaries are explicitly detected.
- No prior information e.g. color, texture, or height of objects in the environment is required.
- No other information besides boundary lines is required for the map. This implies that our method does not care whether a region in the map is a road or a building or anything else.

We evaluated our system through experiments using real-world data sets. We demonstrate that our method enables near real-time, 2D street map-based localization without using prior information.

II. PROPOSED METHOD

A. Model of the World and Representation of Maps

Our method uses 2D street maps consisting of boundary lines as shown in Fig. 2 left. We extract segments of boundary lines and closed regions and assign them unique IDs as shown in Fig. 2 right. In this paper, we call these IDs *segment IDs*.

The procedure is motivated by our model of the world (Fig. 3). We consider the problem of estimating the position of a robot when it observes sensor data from n places in the environment. We denote features extracted from the sensor data by $\{x_1, \dots, x_n\}$. Localization can be achieved by determining the locations from where the sensor data arises. If we have information regarding the relationship between X and map segment Y (e.g. $P(X|Y)$), the solution is

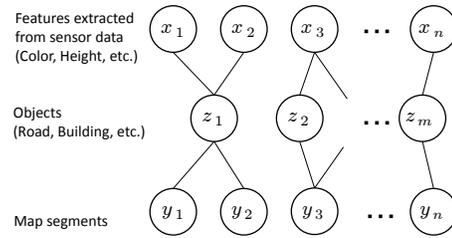


Fig. 3. Model of the world. Sensor data and street map have implicit dependency through objects in the world.

straightforward. The method described in [13] finds implicit relationship between X and Y under assumptions that some knowledge (i.e. $P(Z|Y)$ and $P(X|Z)$) are given in advance. However, these probabilities vary depending on the places. For example, appearance of *road* varies from country to country; hence, $P(X|Z = road)$ also varies.

Therefore, we take a completely different approach, where we find the dependency between X and Y . We assume that segments in maps Y are defined according to the type of objects Z (e.g. road, buildings, etc.); i.e., it is expected that the same type of objects exist in areas with the same segment ID. Naturally, features from sensor data X depends on objects Z being observed; hence, there are latent dependencies between X and Y . Therefore, if the correspondence between X and Y is correct, certain dependency between X and Y should be observed.

B. Mutual Information as a Localization Metric

We propose to employ an MI-based dependence measure to correlate sensor data and a map. MI has been conventionally used for registration of multi-modal images [14]. A standard approach such as Normalized Cross Correlation (NCC) can be used to register normal (unimodal) images as they typically have *linear* dependency between their intensity values. Maximization of NCC leads to maximization of linear dependence between pixel values in two images. However, this is not always true when the two images are of different modalities (e.g. MRI and CT in medical imaging). Registration is performed by maximizing MI, which corresponds to maximizing *non-linear* dependence between pixel intensity values. Our method applies this idea to mobile robot localization, where we match sensor data to a map by finding *non-linear* dependency between them.

The procedure of our method is as follows. First, the robot collects sensor data to extract features from n different places in the environment. We denote the set of d -dimensional features by $\{x_1, \dots, x_n\}$. It is assumed that relative positions between the robot and the sensor data are known. Second, correspondence between the extracted features and a map is made using the robot pose hypothesis w . Segment ID that corresponds to x_i is denoted by y_i . Finally, we obtain n paired input data $\{(x_1, y_1^{(w)}), \dots, (x_n, y_n^{(w)})\}$.

Features and map segment IDs are denoted by random variables X and $Y^{(w)}$. By assuming that the input data is randomly generated from a joint probability distribution $P(X, Y^{(w)})$, we search for w that maximizes a function f

that measures dependence, to localize a robot.

$$\hat{w} = \underset{w}{\operatorname{argmax}} f(X, Y^{(w)}) \quad (1)$$

We propose to use SMI, a variant of MI, as the measure f [17]. The definition and estimation of SMI are detailed in Section III.

In this paper, we focus on 2D position tracking; we estimate the 2D position and orientation of the robot during navigation under the assumption that the initial position of the robot is known (with some errors).

III. ESTIMATION OF SQUARED-LOSS MUTUAL INFORMATION

A. Squared-loss Mutual Information

Mutual Information (MI) is a measure of statistical dependence between random variables and has been used to find dependencies in data. Medical image registration is one of the well-known applications of MI [14]. In our method, we employ Squared-loss Mutual Information (SMI), instead of the original MI, to measure the dependence between sensor data and a map. The definitions of MI and SMI are shown in Eq. (2) and (3).

$$\operatorname{MI}(X, Y) := \iint p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dx dy, \quad (2)$$

$$\operatorname{SMI}(X, Y) := \frac{1}{2} \iint p(x)p(y) \left(\frac{p(x, y)}{p(x)p(y)} - 1 \right)^2 dx dy. \quad (3)$$

SMI can be used as an alternative dependence measure; random variables X and Y are statistically independent if and only if $\operatorname{SMI}(X, Y)$ is equal to zero. SMI has several advantages over original MI [17]:

- Better robustness against outliers. Because SMI does not have logarithm
- Computationally efficient estimator is known

B. Estimating SMI Using LSMI

Our method employs a recent technique on SMI estimation (LSMI [18]). We briefly review the method.

A naive way to estimate Eq. (3) is to estimate all components $p(x)$, $p(y)$, and $p(x, y)$ individually and substitute them in Eq. (3). However, this approach would result in large estimation errors because estimation errors in each component can be combined and magnified. In LSMI, we estimate density ratio

$$r(x, y) = \frac{p(x, y)}{p(x)p(y)} \quad (4)$$

directly (without estimating each component) to improve accuracy and robustness. The density ratio r is modeled by a multiplicative kernel model as shown in Eq. (5).

$$r_{\Theta}(x, y) := \sum_{l=1}^b \sum_{l'=1}^b \Theta_{l,l'} K(x, \tilde{x}_l) L(y, \tilde{y}_{l'}^{(w)}) \quad (5)$$

The parameter to estimate Θ is a matrix of $b \times b$ ($b \leq n$) and each element in it is denoted by $\Theta_{l,l'}$. Kernel centers $(\tilde{x}_1, \dots, \tilde{x}_b$ and $\tilde{y}_1^{(w)}, \dots, \tilde{y}_b^{(w)})$ are randomly chosen from the input data. We use the Gaussian kernel for K and the delta kernel for L as shown below.

$$K(x, \tilde{x}) = \exp\left(-\frac{\|x - \tilde{x}\|^2}{2\sigma^2}\right) \quad (6)$$

$$L(y, \tilde{y}) = \begin{cases} 1, & \text{if } y = \tilde{y} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

The parameter $\hat{\Theta}$ is learned to minimize the squared error in Eq. (9).

$$\min_{\Theta} J(\Theta), \quad (8)$$

$$J(\Theta) := \iint (r_{\Theta}(x, y) - r(x, y))^2 p(x)p(y) dx dy. \quad (9)$$

The minimizer is obtained by solving the following discrete Sylvester equation

$$\frac{1}{n^2} \mathbf{K}^T \mathbf{K} \hat{\Theta} \mathbf{L}^T \mathbf{L} + \lambda \hat{\Theta} = \frac{1}{n} \mathbf{K}^T \mathbf{L}. \quad (10)$$

Here, we approximated the expectations by empirical averages and an ℓ_2 regularization term was added to avoid overfitting. The dimensions of matrices \mathbf{K} and \mathbf{L} are $n \times b$, and their elements are given by $\mathbf{K}_{i,l} = K(x_i, \tilde{x}_l)$, $\mathbf{L}_{i,l} = L(y_i, \tilde{y}_l^{(w)})$. Gaussian kernel width σ and regularization parameter λ can be determined by cross-validation. Finally, the estimated SMI is given as follows

$$\operatorname{LSMI} = \frac{1}{2n} \operatorname{tr}(\mathbf{K} \hat{\Theta} \mathbf{L}^T) - \frac{1}{2}. \quad (11)$$

We use the K -fold cross-validation ($K = 5$ in our experiments) to determine σ and λ . The input data is split into K disjoint sets. Each set is in turn used as test data and the rest of the data is used to estimate parameter Θ . The squared error (Eq. (9)) is evaluated for each set. Hyperparameters that minimize the average error are chosen from several candidates.

IV. APPLICATION

In this section, we describe how to apply our method to implement a localization system. The first two subsections detail how to handle different sensor configurations to associate observations with a map. The last describes how our method can be used for position tracking.

A. Local Grid Map Matching

This implementation employs a 2D grid map of observed features. The robot is assumed to have a 3D sensor such as a laser scanner and a stereo camera. A 3D local map of the environment around the robot is constructed and features such as height and color are extracted. The ground plane is divided into grid cells and the extracted features are projected onto them to generate a 2D grid map. We denote the features contained in the i -th cell of the grid map by x_i . Then, correspondence between the cells and street map is made by translating the local grid map using the current

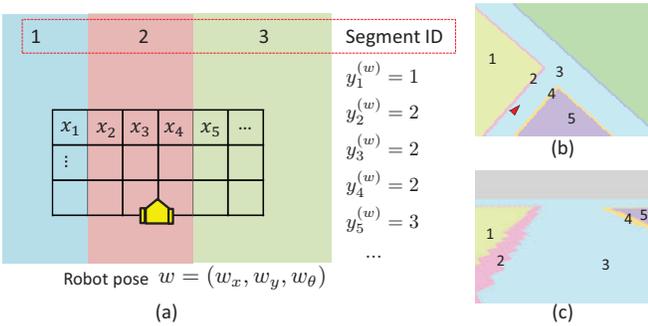


Fig. 4. Examples of associating sensor data with street maps. (a) features stored in a local grid map are matched with a map. (b) segment IDs in the map. (c) segment IDs projected onto the image plane.

robot pose hypothesis $w = (w_x, w_y, w_\theta)$. The segment ID that corresponds to x_i is denoted by $y_i^{(w)}$ (see Fig. 4 (a)). The input data pairs $\{(x_1, y_1^{(w)}), \dots, (x_n, y_n^{(w)})\}$ are obtained through this process. We search for the optimal w by Eq. (1).

B. Localization from a Single Image

Localization can also be achieved using a single image. In this implementation, we use a monocular camera mounted on the robot and assume that the relative pose of the camera with respect to the robot is known.

Features such as color and texture are extracted from the image and handled in a pixel-wise manner. The set of features for the i -th pixel is denoted by x_i . The correspondence between segment IDs and pixels is made by projecting the street map onto the image plane (Fig. 4 (b) and (c)). After the input data pairs are obtained, we search for the optimal w .

C. Integrating into a Particle Filter

1) *Formulation:* Particle filters that fuse observations and the robot’s motion estimation are widely used for robust localization [19] [2]. In this section we describe an implementation of the particle filter using SMI.

In each prediction step of the particle filter, the robot estimates its motion u_t since the previous time step (e.g. by odometry). Particles are drawn from a proposal distribution

$$w_i \sim P(w^t | w_i^{t-1}, u_t). \quad (12)$$

In the update steps of the particle filter, particles are resampled according to the weight proportional to the observation likelihood as follows

$$W_i \propto P(X | w_i, M). \quad (13)$$

The map is denoted by M . We calculate SMI for each particle and interpret the value as the observation likelihood

$$P(X | w_i, M) = \text{SMI}(X, Y_{w_i}). \quad (14)$$

2) *Computationally Efficient Implementation:* Calculation of LSMI itself is computationally efficient; however, the process of cross-validation consumes a considerable amount of time. This can be a hindrance to implement a real-time system. In our implementation, we reduce the number of cross-validation procedures. We augment the state vector of a particle s by appending hyper parameters σ and λ .

$$s_i = (w_i, \sigma_i, \lambda_i). \quad (15)$$

The very first update step, we perform cross-validation for all particles and store the chosen parameters to their state vector. In the subsequent update, we define a *reduction rate* and randomly choose particles for which we skip cross-validation. For example, if the reduction rate is set to 70%, we perform cross-validation for only 30% of particles and we reuse the previously chosen hyper parameters for the rest of particles. By using this trick, we can uniformly reduce the processing time by changing the reduction rate while keeping the each update time constant.

V. EVALUATION

In this section we demonstrate the performance of our method by four experiments using three data sets collected in urban environments. Experiments shown in this paper are summarized in Tab. II.

A. 2-DoF Localization Using Single Images

First, we evaluated the effectiveness of SMI as a measure of localization. 37 images of urban scene (32 from sidewalks and 5 from roadways) were used for evaluation. The features shown in Tab. I were extracted from images and used as the input data. The ground truth positions were given manually.

The street map we used was generated from Google Map¹. A screen capture of Google Map was saved as an image and closed regions are extracted manually. Then, boundary lines are extracted by an edge detection algorithm. Finally, unique IDs are assigned by the flood-fill algorithm [20]. Fig. 5 illustrates the procedure.

We conducted 2-DoF (lateral position and orientation with respect to road) localization using single images. The reason we did not perform 3-DoF localization is that 2D street maps are often non-discriminative in longitudinal direction and longitudinal position can be easily estimated by robot’s motion estimation. A grid search was used to find the position that maximizes Eq. (1). The search ranges were -2m to $+2\text{m}$ with interval of 0.2m for lateral offset and -20deg to $+20\text{deg}$ with interval of 2deg for orientation.

Since no existing method is known to be able to perform localization in the same setup, we implemented a comparative method which is based on road boundary matching. Road boundaries were detected by the road detection method of Kong *et al.*² [21], which uses vanishing point detection

¹Other maps such as OpenStreetMap can also be used. In this paper we employed Google Map because, for our target area, it provided much more accurate sidewalk information compared to OpenStreetMap.

²We used the code downloaded from <http://web.mit.edu/huikong/www/code.html>.

TABLE I
FEATURES USED IN SINGLE IMAGE
LOCALIZATION EXPERIMENTS

Category	Feature
Color	R, G, B
Gradient	Sobel magnitude, orientation

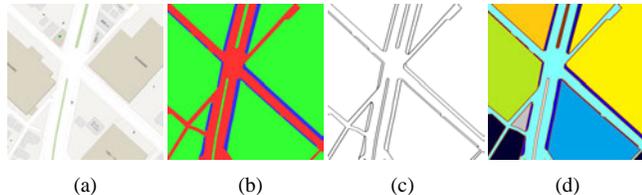


Fig. 5. Illustration of map generation procedure. (a) Original map image captured from Google Map. (b) Manually extracted closed regions. (c) Boundary lines are detected by an edge-detection algorithm. (d) Unique IDs are assigned by the flood-fill algorithm [20]. Different colors indicate different IDs.

and does not use any prior information about color or texture of the road. The detected road boundaries were matched with lines in the map. Robot positions where boundary lines match best were found by the same grid search mentioned above.

The results are summarized in Fig. 6. Our method outperformed the road matching method in terms of localization accuracy. The average position and orientation errors were 0.64m and 6.5deg. The resolution of the map was approximately 23.5cm per pixel and the position error appears as 2.7 pixels in the map. We consider the localization accuracy reasonable because the localization error includes errors in the map and the street map was not pixel-wise accurate.

Exemplary results are shown in Fig. 7, with LSMI plots against position and orientation errors (rightmost column). Successful examples for both methods are (a) and (b). The LSMI plot (a) shows a clear peak around the ground truth position. While the road detection method failed on (c, d, e) because of the confusing salient edges in the images, our method successfully localized them. Braille blocks caused several localization failures. Two peaks can be seen in the plot of (f); the highest peak appeared at the ground truth and the second highest one matched the Braille blocks. Some of the localization errors were caused by map errors. An example of road width errors can be seen in (g). Both road detection and our method failed on (h) in which road boundaries were not clear and the feature differences were very small between the roadway, the sidewalk and the parking lot. A challenging example on map errors is shown in (i). The left road boundary in the map appears curved, but it is actually not curved. Another challenge is coping with occlusions like (j); our method failed because the blue tarp covering the road had similar color to the vehicle which appears on the right hand side of the image.

B. Position Tracking Using a Particle Filter

We conducted full (3-DoF) localization experiments. We integrated our method into a particle filter and conducted

TABLE II
LIST OF EXPERIMENTS, METHODS AND DATA SETS

Section	Type of experiments	Methods applied	Data set
V-A	Single image localization	IV-B	37 images
V-B	Position tracking	IV-A, IV-C	Velodyne scans, 150m odometry
V-C.1	Single image localization	IV-B	4,556 images, 4.1km odometry
V-C.2	Position tracking	IV-B, IV-C	

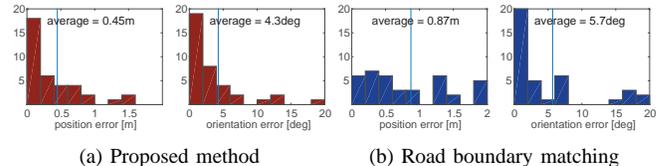


Fig. 6. Histograms of position and orientation errors from single image localization experiments using 37 images.

position tracking experiments. We collected log data of odometry, laser measurements and Real Time Kinematic (RTK) GPS in a 150m pathway in Narashino, Japan, by manually navigating a wheel-chair robot (Fig. 8 left). Localization experiments were conducted off-line using the collected data.

To correlate sensor data with the map, local grid maps were generated at 1 Hz by dividing the ground plane around the robot into square cells. The laser measurements were accumulated for one second to generate a 3D point cloud. The points in the cloud were projected onto the ground, and stored in the grid cells. The size of the grid cells was matched with the map resolution (23.5cm). For each cell in the grid map c_1, \dots, c_n , the minimal value and the variance of the point heights were calculated and used as 2-dimensional input features x_1, \dots, x_n .

The result of position tracking is shown in Fig. 8 right. It can be seen that our method successfully corrected the odometry errors and kept track of the robot position. We evaluated the localization accuracy using GPS as the ground truth. The root mean square (RMS) error of the estimated trajectory was 0.59m (approximately 2.5 pixels in the map).

We compared the processing time of particle filter update and the localization accuracy with different reduction rates (cf. Section IV-C.2). Fig. 9 summarizes the results. The estimated trajectory with 80% of the reduction rate is also shown in Fig. 8. By reducing 80% of cross-validation procedures, the average processing time decreased from 2.9 seconds to 0.7 seconds, without significant degradation in localization accuracy. Our implementation was written in MATLAB and the time was measured on a laptop computer with a 2.4GHz Core i7-5500U CPU. The number of particles employed was 50, and the number of kernel centers used to estimate SMI was 40.

C. Experiments Using a Large-scale Data Set

For a large-scale evaluation of our method, we collected another data set by manually navigating the wheel-chair robot in an urban area in Narashino, Japan. During the 5.8km navigation, 4,556 images (captured with one second interval),

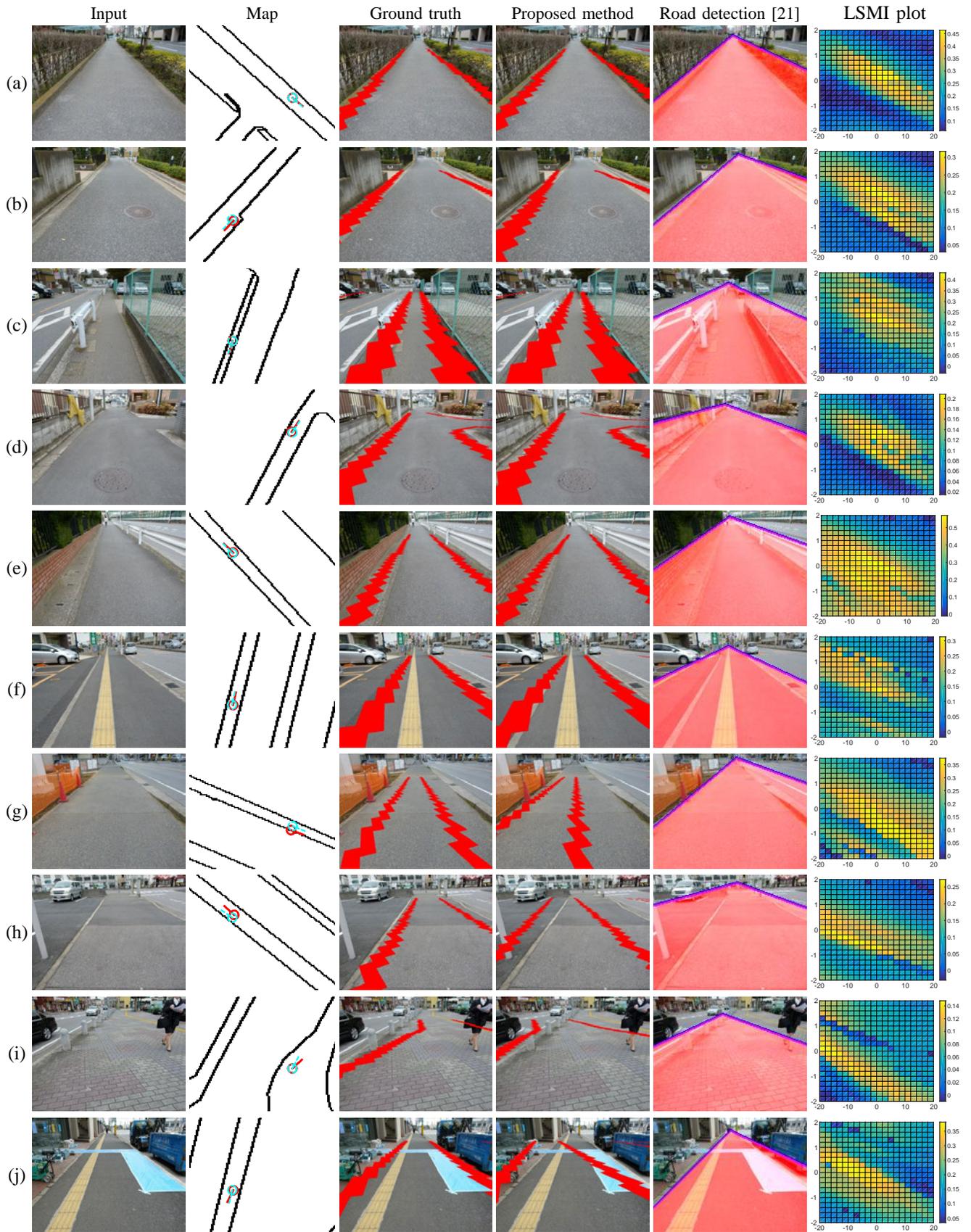


Fig. 7. Exemplary results of localization experiments. Localization results are shown by superimposing the road boundaries to the images. In the maps the estimated positions and the ground truth positions are indicated in red and blue, respectively. In the comparative method, detected road boundaries were matched with the map.



Fig. 8. Left: robot used to collect data. It is equipped with a Point Grey Flea3 camera, a Velodyne HDL-32e laser scanner, a gyro and a GPS receiver (Ladybug was not used). Right: result of position tracking experiments. The red trajectory shows the result of our method without reduction of cross-validation procedures. The green trajectory shows the result with 80% of cross-validation reduction.

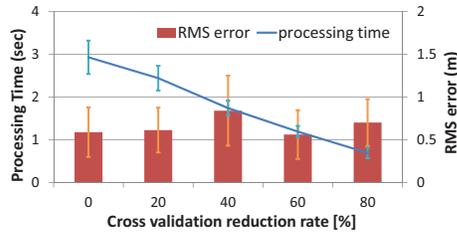


Fig. 9. Comparison of the processing time of the update and the localization error with different cross validation reduction rates. Error bars indicate standard deviations.

odometry and RTK-GPS logs were collected. Images were captured by a Point Grey Flea3 camera with a fish-eye lens. We conducted two kinds of experiments, single image localization and positing tracking, using the data set.

1) *Single Image Localization*: We conducted experiments by the same procedure as described in Section V-A, except that we used GPS for the ground truth. Since the GPS measurements were not always accurate, we manually extracted 575 images with good GPS accuracy for evaluation. The results of single image localization experiments are summarized in Fig. 10, and exemplary results are shown in Fig. 11. The number of images localized within 1m and 10deg of errors was 388 (67%) with the proposed method, and 211 (34%) with the road boundary matching method. The accuracy was slightly worse than the results in Section V-A. The adverse illumination conditions (see Fig. 11 (b)) would be the cause of the performance degradation.

2) *Position Tracking*: We implemented a particle filter that fuses odometry and image measurements. The particle filter update was executed for every image observation (one second interval). Fig. 12 shows the result of the position tracking. Our proposed method successfully kept track of the robot during the 4.1km of run from the start point to the point (A) shown in the figure. Unfortunately, the odometry log ended at the point because of a system malfunction. The RMS error of the estimated trajectory against the GPS log was 1.6m (note that the GPS log also contained some errors).

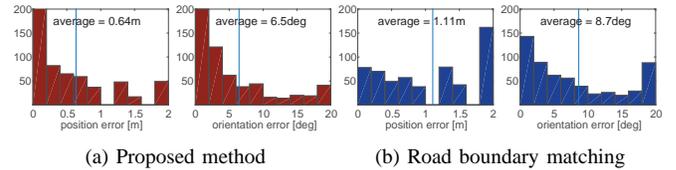


Fig. 10. Histograms of position and orientation errors from single image localization experiments using the large scale data set. 575 images were used for evaluation.

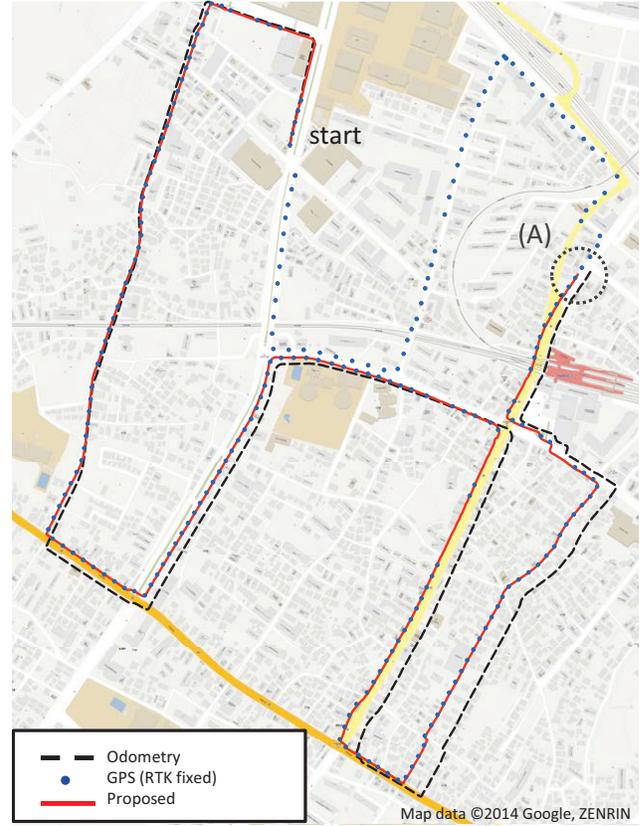


Fig. 12. Position tracking result using the large-scale data set. Our method successfully tracked the robot position from the start to point (A) where the the odometry log ended (4.1km in the distance).

VI. CONCLUSIONS

In this paper, we presented a novel method for street map-based localization. Matching sensor data with a street map arises as an issue in environments where road boundaries cannot be easily detected. Our method overcomes the issue using a statistical dependence measure. Localization is performed through maximization of Squared-loss Mutual Information. Unlike previous methods, our method requires no prior information about the environment. The validity of the proposed method is supported by experiments using real-world image and laser data sets. It is also shown that our implementation is efficient enough for autonomous navigation use.

Further research could include feature extraction. In this paper, we have used rather simple features; however, more sophisticated set of features should be explored for robust localization. Another significant concern is handling of map errors. We expect that small errors in a map can be addressed

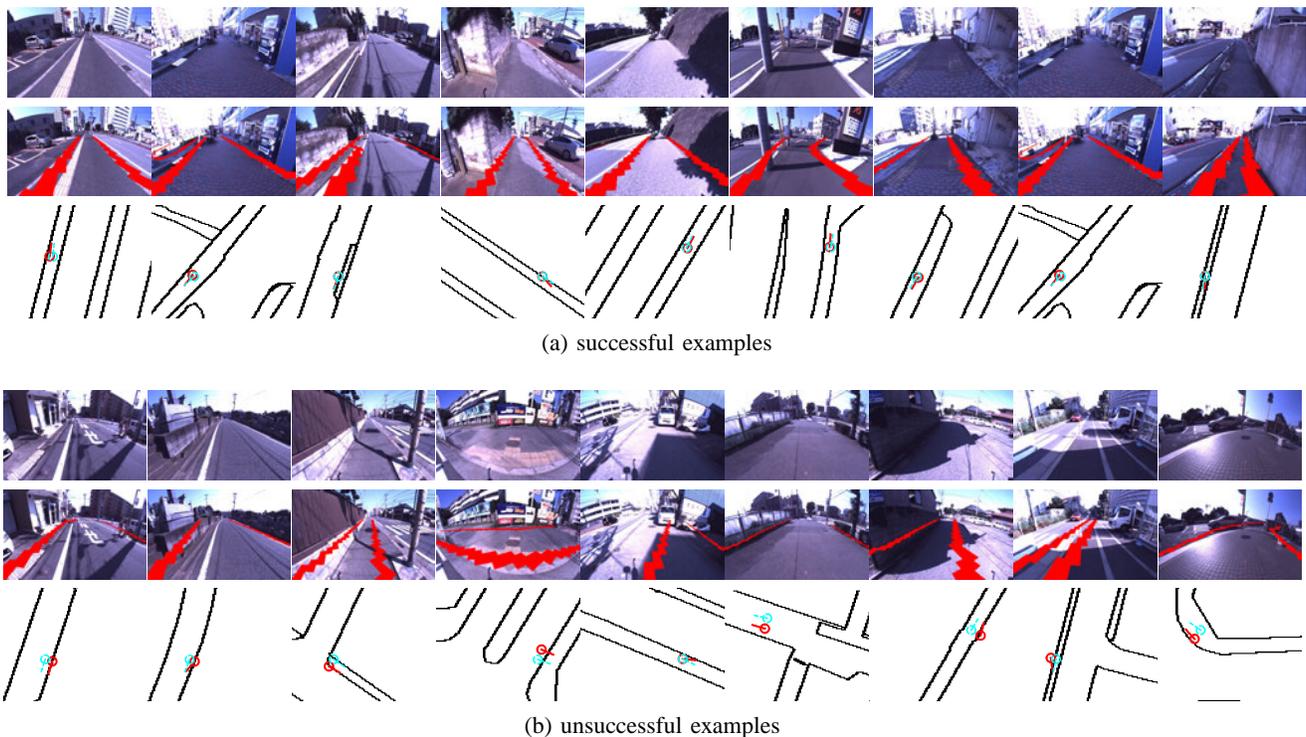


Fig. 11. Exemplary results of single image localization experiments using the large image data set. The first rows show input images and the second rows show localization results by superimposing the road boundaries. The third rows show estimation results (red) and ground truth positions (blue).

by deforming local grid maps.

Since our method is general and not restricted on the road, it should be able to be applied to localization of other types of robots such as flying robots. While our method still has many issues to be addressed, we believe it is a significant step towards street map-based localization.

REFERENCES

- [1] Y. M. Saiki, E. Takeuchi, A. Carballo, W. Tokunaga, H. Kuniyoshi, A. Aburadani, A. Hirose, Y. Nagasaka, Y. Suzuki, and T. Tsubouchi, "1km autonomous robot navigation on outdoor pedestrian paths "running the Tsukuba challenge 2007", in *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, 2008, pp. 219–225.
- [2] R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard, "A navigation system for robots operating in crowded urban environments," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013, pp. 3225–3232.
- [3] M. Adams, S. Zhang, and L. Xie, "Particle filter based outdoor robot localization using natural features extracted from laser scanners," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2004, pp. 1493–1498.
- [4] E. Royer, M. Lhuillier, M. Dhome, and J.-M. Lavest, "Monocular vision for mobile robot localization and autonomous navigation," *Int. J. of Computer Vision*, vol. 74, no. 3, pp. 237–260, 2007.
- [5] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2010, pp. 4372–4378.
- [6] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk, "On map-matching vehicle tracking data," in *Proceedings of the 31st International Conference on Very Large Data Bases, ser. VLDB '05*, 2005, pp. 853–864.
- [7] I. Parra Alonso, D. Fernandez Llorca, M. Gavilan, S. Alvarez Pardo, M. Garcia-Garrido, L. Vlacic, and M. Sotelo, "Accurate global localization using visual odometry and digital maps on urban environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1535–1545, 2012.
- [8] M. A. Brubaker, A. Gejger, and R. Urtasun, "Lost! leveraging the crowd for probabilistic visual self-localization," in *Proc. of IEEE Computer Society Conference on Vision & Pattern Recognition (CVPR)*, 2013.
- [9] G. Floros, B. van der Zander, and B. Leibe, "OpenStreetSLAM: Global vehicle localization using OpenStreetMaps," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013, pp. 1054–1059.
- [10] F. Chausse, J. Laneurit, and R. Chapuis, "Vehicle localization on a digital map using particles filtering," in *Proc. of the IEEE Int. Conf. on Intelligent Vehicles Symposium*, 2005, pp. 243–248.
- [11] Y. Morales, T. Tsubouchi, and S. Yuta, "Vehicle 3D localization in mountainous woodland environments," in *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, 2009, pp. 3588–3594.
- [12] M. Hentschel and B. Wagner, "Autonomous robot navigation based on OpenStreetMap geodata," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems (ITSC)*, 2010, pp. 1645–1650.
- [13] K. Irie and M. Tomono, "Localization and road boundary recognition in urban environments using digital street maps," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012, pp. 4493–4499.
- [14] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever, "Mutual information based registration of medical images: a survey," *IEEE Trans. on Medical Imaging*, vol. 22, no. 8, pp. 986–1004, 2003.
- [15] A. Alempijevic, S. Kodagoda, and G. Dissanayake, "Cross-modal localization through mutual information," in *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, 2009, pp. 5597–5602.
- [16] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Toward mutual information based place recognition," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014, pp. 3185–3192.
- [17] M. Sugiyama, "Machine learning with squared-loss mutual information," *Entropy*, vol. 15, no. 1, pp. 80–112, 2013.
- [18] T. Sakai and M. Sugiyama, "Computationally efficient estimation of squared-loss mutual information with multiplicative kernel models," *IEICE Trans. on Information and Systems*, vol. E97-D, no. 4, pp. 968–971, 2014.
- [19] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2000.
- [20] J. R. Shaw, "QuickFill: An efficient flood fill algorithm," <http://www.codeproject.com/Articles/6017/QuickFill-An-efficient-flood-fill-algorithm>, 2004.
- [21] H. Kong, J. Audibert, and J. Ponce, "General road detection from a single image," *IEEE Trans. on Image Processing*, vol. 19, no. 8, pp. 2211–2220, 2010.