Utilizing WiFi Signals for Improving SLAM and Person Localization*

Taku Kudo and Jun Miura

Abstract— **This paper describes the use of WiFi signals for improving SLAM and person localization problems. Loop closing is the most important step in large-scale mappings and many previous methods rely on image-based feature matching. Such methods are, however, usually costly and tend to sensitive to illumination variations and the robot heading. We therefore propose a new loop closure detection method using WiFi fingerprints. Since WiFi signals are sometimes quite uncertain, we do matching not a pair of poses but a pair of pose sequences for improving the loop closing performance. We then use the WiFi-recorded map to localizing a person with a smartphone. We develop a particle filter-based method and apply it to a robot call system. Experimental results show the effectiveness of the proposed methods.**

I. Introduction

SLAM (Simultaneous Localization And Mapping) is one of the most important technologies for mobile robots working in unknown environments [1]. Among many SLAM methods, LIDAR-based 2D mapping is the most common, and loop closing is always a problem in a large scale SLAM. This paper deals with the problem in a LIDAR-based 2D SLAM.

Loop closing is the most important step in large-scale mappings There are various ways to solving the loop closing problem such as LIDAR-based (e.g., [2]) and image-based (e.g., [3]). These approaches work well when applied to the environment and the condition where appropriate features are sufficiently observed. If not due to, for example, an bad illumination condition, they may fail to correctly close loops.

As wireless connection services become popular in public spaces, using WiFi signals could provide extra information to robot and person localization. Several methods have been proposed for improving localization accuracy (e.g., [4]) but not applied to improving loop closing.

In this paper, we propose a new way of using WiFi signals for improving a LIDAR-based SLAM. Since WiFi signals are sometimes quite uncertain, we do matching not a pair of poses but a pair of pose sequences for improving the loop closing performance. We also develop a WiFi-based person localization method. Since the generated map includes recorded WiFi information, the method utilizes it in a particle filter-based localization. As an application of the combination of WiFi-supported SLAM and person localization, we develop a robot call system by which the user can easily call the robot from anywhere. We implemented the proposed methods and evaluated through mapping and robot call experiments.

*This work is in part supported by JSPS KAKENHI Grant Numbers 25280093 and 17H01799.

The authors are with the Department of Computer Science and Engineering, Toyohashi University of Technology, Toyohashi, Japan.

The rest of the paper is organized as follows. Section II describes related works on loop closing and person localization. Sec. III describes a WiFi-augmented loop closing and mapping results. Sec. IV describes a WiFi-based person localization and its application to a robot call system. Sec. V concludes the paper and discusses future work.

II. Related Work

A. Loop closing

There are several approaches to solving the loop closing problem. Blanco et al. [2] developed a method of optimizing a map with loops using LIDAR and odometry information. Their method, HMT-SLAM, divides a map into several areas and compares the shape of grid maps of those areas to optimize the whole map while keeping its topological structure. The method works well in environments with a rich shape variety, but might be weak in a simple environment such as the one composed of long straight corridors. Other LIDAR and odometry-based methods suffer from similar drawbacks.

Image-based loop closure detection [5], [6], [7] is also popular. These methods represent locations with an image feature such as BOVW (bag of visual words) [8], and compare the current image with past ones to find loop closures. Calculation of image features is relatively costly, although several attempts exist to reduce the cost by limiting the number of image comparison [3], [9], [10]. Moreover, image-based methods sometimes suffer from a limited field of view and sensitivity to illumination changes.

B. WiFi-based localization

Huang et al. [4] developed a method of estimating a user trajectory by combining odometry-based ego-motion with a WiFi signal similarity in a pose graph optimization framework. Ferris et al. [11] proposed to use Gaussian Process Latent Variable Model (GP-LVM) to represent WiFi signal distributions in a mapping area. They optimize the marginalized latent positions by minimizing the negative log-likelihood of GP-LVM that was composed of positions and measured WiFi signal strengths. These works exhibit a good performance but not used for increasing map quality in a robotic LIDAR-based mapping.

Ito [12] proposed a mobile localization method using WiFi signals with a Gaussian process particle filter. This method is effective especially when a dense WiFi data set is obtained.

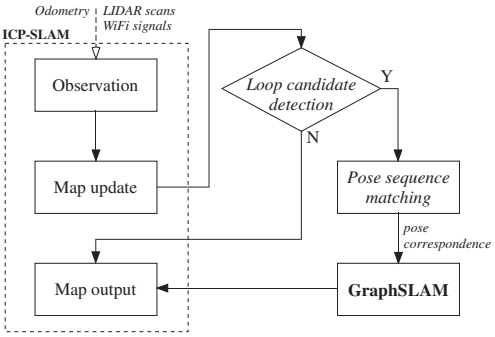


Fig. 1. Flow of the proposed method.

III. WiFi-augmented Loop Closing

A. Overview of the SLAM method

This paper proposes a new loop closing method using WiFi signals. We use ICP-SLAM [13] as a based SLAM and a GraphSLAM algorithm [14] for pose-graph optimization after detecting loop closures.

Fig. 1 shows the proposed SLAM framework. The inputs are odometry data, LIDAR scans, and WiFi signals. ICP-SLAM updates the map when the robot moves by a certain distance or a certain angle. *Loop Closure Detection* step checks if a loop candidate exists based on the robot odometry and the WiFi signal similarity. If detected, *Pose Sequence Matching* step verifies it. If verified, the map is updated by GraphSLAM with newly added loops.

We use MRPT (Mobile Robot Programming Toolkit) [15] implementations of the SLAM algorithms, but the proposed loop closure method can be applied to any implementation.

B. Loop Candidate Detection

1) Robot pose uncertainty: Loop candidate detection is performed for pairs of the current and a past pose. The first step is to find possible pairs using a pose uncertainty estimate, calculated by accumulating odometry errors.

2) WiFi signal handling: WiFi signal strengths are measured by a common wireless LAN receiver device and Native WiFi API [16], at every two to three seconds. At each map update, the latest value is recorded with the robot pose. Native WiFi uses the following expression to define the signal quality SQ:

$$SQ = 2 \cdot (RSSI + 100), \tag{1}$$

where $RSSI$ indicates the received signal strength indicator, expressed in the range [0: 100] in Decibel-milliwatt.

The source of a signal is determined by BSSID (Basic Service Set Identifier), uniquely assigned to each AP (access point). All signal strength data observed at all poses are compiled in a two-dimensional signal-pose matrix S, which has the dimension of $T \times N$ where T is the number of robot poses and N is the number of APs detected so far. Both dimensions increase as a new AP is found or a new observation is performed.

We do not use raw signal data because they are usually very noisy. Instead, we apply a temporal smoothing to them. Let $\boldsymbol{s}_n$ be the nth column vector of S, and SA be the temporally-smoothed WiFi signal matrix. Each element SA_{tn} is given by:

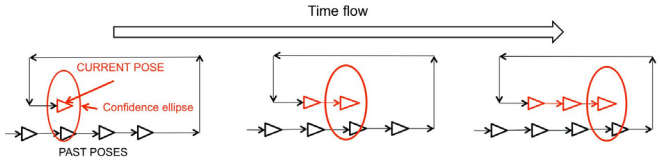


Fig. 2. Making a current sequence.

$$SA_{tn} = \boldsymbol{\beta}_t^T \boldsymbol{s}_n, \tag{2}$$

where $\boldsymbol{\beta}_t$ is a weight vector for the smoothing and its elements are given by [17]:

$$[\boldsymbol{\beta}_t]_i \propto \exp\left\{-\frac{1}{2\tau^2} dist\left(Pose_t - Pose_i\right)^2\right\}, \tag{3}$$

$$\sum_{i=0}^{T-1} [\boldsymbol{\beta}_t]_i = 1, \tag{4}$$

where $dist(Pose_t - Pose_i)$ is the cumulative travel distance between the ith and the tth pose; τ is a parameter to control the smoothing, currently set to $1.5\,m$. This weighting is based on a model that WiFi signals are attenuated as the distance between two poses increases in a wall-free environment [17]. We then define the WiFi signal strength similarity Sim_{ij} between the ith and the jth pose as the cosine similarity of the corresponding vectors in SA.

3) Loop candidate detection by constructing a sequence pair: Loop candidate detection is to find a pair of pose sequences, the current one (Current Sequence, CS) and the past one (Past Sequence, PS). Fig. 2 shows the process of CS detection based on the positional correspondence. The process starts when a revisit of a past location is detected using the confidence ellipse (we use 3σ ellipse). It continues until the corresponding pair is no longer detected. Since the first and the last part of a CS could include poses which are likely to have no corresponding poses in the past, the poses inside the first and the last confidence ellipses are deleted. The remaining poses constitutes a CS. Fig. 3 shows an example process of CS detection in a real environment; poses in the confidence ellipses at steps 1 and 3 are deleted.

A PS is then made for a given CS by collecting past poses which have a certain level of WiFi signal similarity with at least one of the poses in the CS. The collected past poses constitutes the PS as shown in Fig. 4.

C. Pose Sequence Matching for Loop Candidate Verification

1) Continuous DP matching between CS and PS: There may sometimes be multiple corresponding candidates from a pose in the CS to those in the PS as shown in Fig. 4. To make a set of *uncrossed* pose correspondence, we adopt DP matching. In our case, since a PS is usually longer than a CS and their starting poses are not always matched, we use continuous DP [18], which is suitable for *spotting* the CS's corresponding part in the PS.

We prepare a cost map as shown in Fig. 5 for the

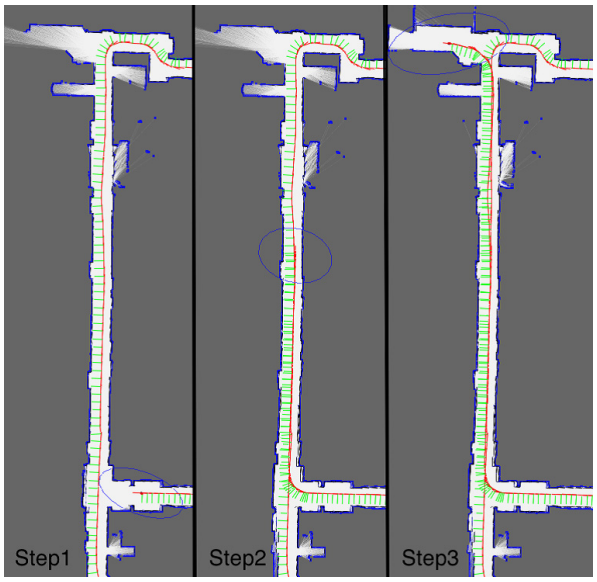


Fig. 3. An example detection of a current sequence.

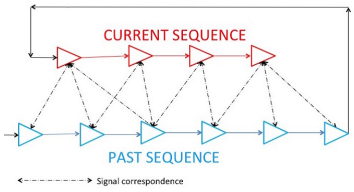


Fig. 4. Making a past sequence for a current sequence.

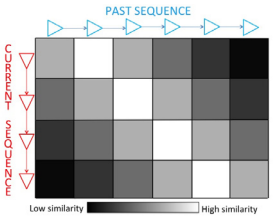


Fig. 5. Cost map.

continuous DP matching. The value of each cell of the cost map is given by the inverse of the similarity value Sim_{ij}. The costs of edges are defined such that horizontal movements become more costly than oblique ones assuming that the robot speed is almost constant in the same area. Choosing a coefficient $\alpha = 1.1$ experimentally, the minimum cost from C_{0*} (any selected node in the top row of cost map C) to C_{ij}, g_{ij}, is calculated by:

$$g_{ij} = \min \begin{cases} g_{(i-1)(j-3)} + C_{i(j-2)} + \alpha C_{i(j-1)} + \alpha C_{ij} \\ g_{(i-1)(j-2)} + C_{i(j-1)} + \alpha C_{ij} \\ g_{(i-1)(j-1)} + C_{ij} \\ g_{(i-2)(j-1)} + C_{(i-1)j} + \alpha C_{ij} \\ g_{(i-3)(j-1)} + C_{(i-2)j} + \alpha C_{(i-1)j} + \alpha C_{ij} \end{cases} \quad (5)$$

We call the matrix composed of g_{ij}'s as the minimum path map G. In the continuous DP matching, we seek the minimum path from G_{0*} to G_{M*} (M is the number of rows (i.e., current poses) in G).

2) ICP-matching for calculating relative poses and for excluding mismatches: We now have two pose sequences with a set of pose-to-pose correspondence. Since all correspondence are just WiFi signal similarity-based, they are usually geometrically different. We thus apply the ICP matching to each of them for calculating their relative pose, and only consistent relative poses are then added to the pose graph as new edges.

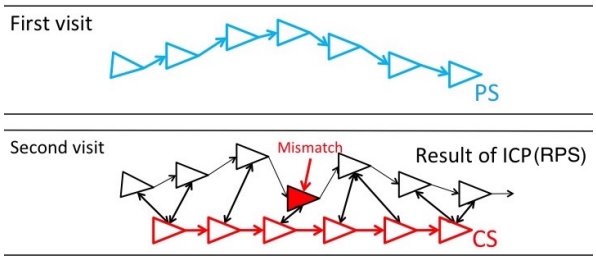


Fig. 6. ICP matching for relative pose calculation.

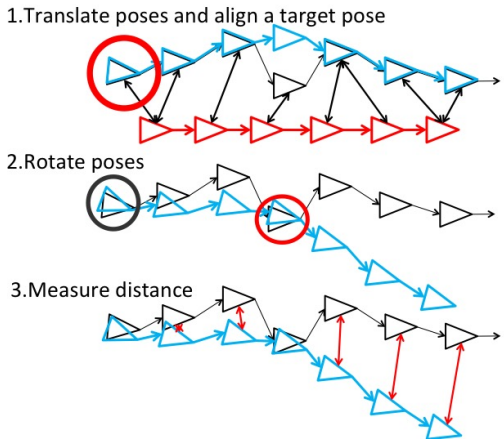


Fig. 7. A mismatch exclusion process. The red, the blue, and the black markers indicate the current sequence (CS), the past sequence (PS), and the relative past sequence (RPS), respectively.

Fig. 6 illustrates an example ICP matching result. In the figure, blue markers indicate the robot pose sequence in the first visit (PS) and red ones indicate those in the second visit (CS). By applying the ICP matching to each pose-to-pose correspondence, we get a relative pose of a past pose with respect to its corresponding current pose, indicated by blacker markers and two-way arrows in the figure. We call this black pose sequence a relative past sequence (RPS). Since the ICP matching does not always give us a correct relative pose, we need to exclude mismatches and obtain a set of consistent relative poses.

Fig. 7 illustrates the process of mismatch exclusion. If all of ICP matching results are correct, two sequences, PS and RPS, almost overlap with each other. If not, some non-overlapping poses appear in RPS. To detect such a case, we evaluate each edge in terms of the consistency with the others. We exhaustively choose pair of correspondence one by one, transform the RPS using the chosen pair (steps 1 and 2 in the figure), and exclude outliers using a statistical test (step 3). All remaining edges (i.e., inliers) are then added to the pose graph for the subsequent pose graph optimization.

D. Mapping experiment

1) Experimental setup: We conducted experiments in the second floor of the buildings in our campus. Our robot is composed of a modified electric wheelchair controlled by a PC (Patrafour by Toyota Motor East Japan) and two LIDARs (Hokuyo UTM-30LX) covering $360\,deg.$, and a GW-300S

Fig. 8. A mapping result with the proposed method.

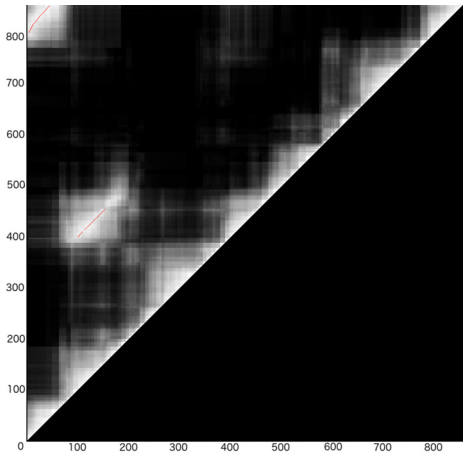
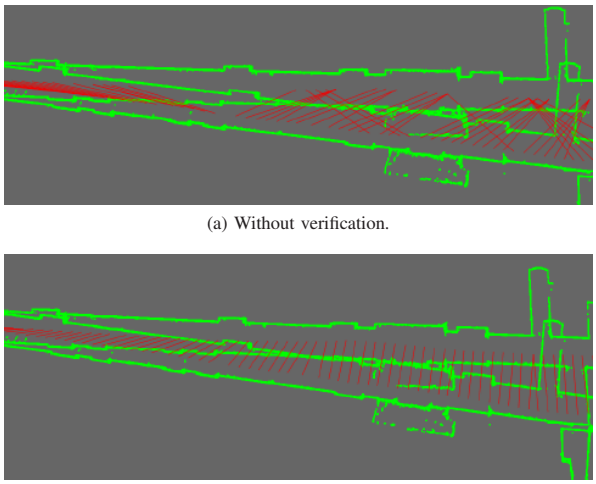


Fig. 9. WiFi signal strength similarity matrix.



(a) Without verification.

(b) With verification.

Fig. 10. Effect of DP matching-based correspondence verification.

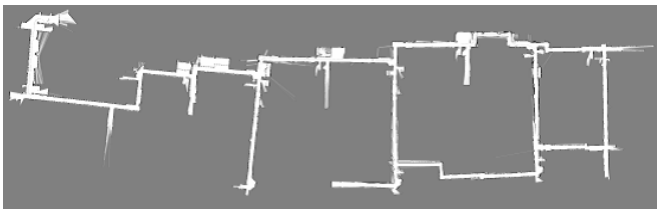


Fig. 11. Mapping of a large indoor environment.

KATANA WiFi receiver module.

2) Mapping results: Fig. 8 shows a mapping result. We can see that two large loops in the environment are correctly closed. Fig. 9 shows the WiFi signal strength similarity matrix (SA in Sec. III-B.2), where brighter values indicate higher similarity. Red dots indicate loop closure sequences detected by our method. We can clearly see two loop closures: one is for the central part (poses around time 150 and those around 450) and the other is the overlap of robot poses near the starting and the end point (poses around time 0 and those around time 800).

Fig. 10 shows an effect of DP matching-based loop verification. Fig. 10(a) shows most-probable correspondence between poses using only the WiFi similarity measure; many crossed corresponding relationships exist. Fig. 10(b) shows the set of final correspondence after taking the verification step, which is consistent enough to be used for loop closing. Fig. 11 shows the result of mapping a larger indoor environment at our university. The size of the environment is about $300\,m \times 50\,m$. The map is basically reasonable to be used for localization, and has correct shapes at regions with loops.

3) Comparison with visual loop closing: Visual features are sometimes very effective in finding correct loop closures. We here choose RTABMap [3] and use its implementation [19] for comparison. We collected images at each robot pose using a 120° wide-angle camera, calculated SIFT features, and constructed a Bag-of-visual-words representation with 400 words for generating input data to RTABMap. Evaluation criteria are the accuracy of loop closure detection and the speed of map update.

a) Accuracy in loop closure detection: We made the ground truth data of pose correspondence at the two loop closing regions in Fig. 8. Fig. 12 shows the error of correspondence in terms of the number of poses apart. Table I summarizes the statistics of the errors. We can see that both methods shows reasonably good accuracy, although our method exhibits a little larger errors. The average pose interval was about $0.5\,m$ in this experiment, and the maximum error (i.e., nine frames) corresponds to about $4.5\,m$ error. The average error is about $75\,cm$ and this is acceptable in spite of the shape feature-scarce environment, especially in a long corridor in the first loop closure region.

b) Calculation time: Fig. 13 shows the comparison results in terms of loop candidate detection and verification.

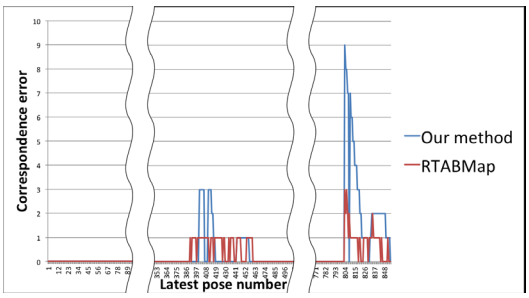


Fig. 12. Correspondence error in loop closure regions.

TABLE I

STATISTICS OF CORRESPONDENCE ERRORS.

Methods	Average	Std.	Max.
Our method	1.461	2.788	9
RTABMap	1.135	0.502	4

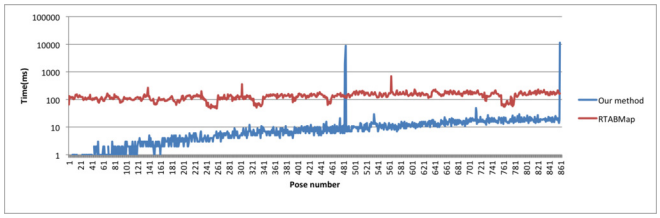


Fig. 13. Comparison with RTABMap in terms of calculation time.

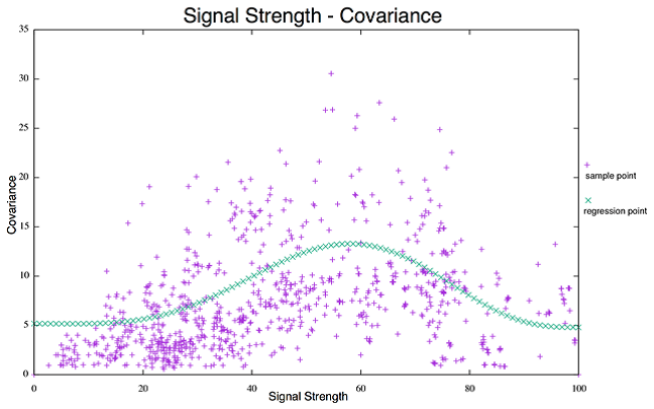


Fig. 14. Strength vs. standard deviation relationship of WiFi signals. Purple plots are measured value and green ones are fitted ones.

Fig. 13(a) shows the calculation time at each map update. RTABMap constantly takes about $100\,ms$ for one loop closure detection, whereas our method takes only about $10\,ms$. Two peaks in the data of our method indicate the time point where the pose sequence verification including ICP matching for estimating relative poses was performed.

IV. WIFI-BASED PERSON LOCALIZATION

A. Predicting WiFi signal strength using Gaussian Processes

The location of a person in a map is useful for a robot to move to and provide the service to the user. Since our map has a set of WiFi signal data recorded with the corresponding robot poses, we would like to utilize it for person localization. We adopt Ito's approach [12] which approximates the distribution of WiFi fingerprints using Gaussian Process (GP).

The set of WiFi data used for GP modeling is given by:

$$D = \{(\boldsymbol{p}_0, \boldsymbol{s}_0), (\boldsymbol{p}_1, \boldsymbol{s}_1), (\boldsymbol{p}_2, \boldsymbol{s}_2), \cdots, (\boldsymbol{p}_N, \boldsymbol{s}_N)\}, \quad (6)$$

where $\boldsymbol{p}_n$ is the nth robot pose and $\boldsymbol{s}_n$ is the WiFi fingerprint obtained at $\boldsymbol{p}_n$. We model the relationship between $\boldsymbol{s}_n$ and $\boldsymbol{p}_n$ using the following form:

$$\boldsymbol{s}_n = f(\boldsymbol{p}_n) + \boldsymbol{\epsilon}, \quad (7)$$

where $\boldsymbol{\epsilon}$ is an additive noise and follows $N(\boldsymbol{0}, \sigma_p^2 I)$. The predicted WiFi fingerprint at position $\boldsymbol{p}$ is described by the mean and the variance. The mean $GP_\mu(\boldsymbol{p}, D)$ is given by:

$$\begin{aligned} GP_\mu(\boldsymbol{p}, D) &= \boldsymbol{k}^T (K + \sigma_p^2 I)^{-1} \boldsymbol{S}, \quad (8) \\ \boldsymbol{S} &= \{\boldsymbol{s}_0, \boldsymbol{s}_1, \cdots, \boldsymbol{s}_N\}^T, \\ k(\boldsymbol{p}_1, \boldsymbol{p}_2) &= \sigma_f^2 \exp\left\{-\frac{1}{2l^2} |\boldsymbol{p}_1 - \boldsymbol{p}_2|^2\right\} \quad (9) \\ \boldsymbol{k} &= \{k(\boldsymbol{p}, \boldsymbol{p}_0), k(\boldsymbol{p}, \boldsymbol{p}_1), \cdots, k(\boldsymbol{p}, \boldsymbol{p}_N)\}^T \\ K_{ij} &= k(\boldsymbol{p}_i, \boldsymbol{p}_j) \end{aligned}$$

The variance $GP_{\sigma^2}(\boldsymbol{p}, D)$ is given by:

$$GP_{\sigma^2}(\boldsymbol{p}, D) = k'(\boldsymbol{p}, \boldsymbol{p}) - \boldsymbol{k'}^T (K' + \sigma_p^2 I)^{-1} \boldsymbol{k'} \quad (10)$$

$$k'(\boldsymbol{p}_1, \boldsymbol{p}_2) = F_s(GP_\mu(\boldsymbol{p}, D)) \exp\left\{-\frac{1}{2l^2} |\boldsymbol{p}_1 - \boldsymbol{p}_2|^2\right\} \quad (11)$$

$$\begin{aligned} \boldsymbol{k'} &= \{k'(\boldsymbol{p}, \boldsymbol{p}_0), k'(\boldsymbol{p}, \boldsymbol{p}_1), \cdots, k'(\boldsymbol{p}, \boldsymbol{p}_N)\}^T \\ K'_{ij} &= k'(\boldsymbol{p}_i, \boldsymbol{p}_j) \end{aligned}$$

In the kernel functions (eqs. (9) and (11)), l is a parameter which indicates an average distance from the robot to a wall in the experimental environment, currently set to $1.5m$, and the norm $|\boldsymbol{p}_1 - \boldsymbol{p}_2|$ is the accumulated distance between the poses.

In [12], a kernel function is used for both mean and variance estimation. In this work, we use a different kernel k' for the variance because we observed that the variance follows a complex function $F_s(GP_\mu(\boldsymbol{p}, D))$ of a signal strength. We measured the actual mean and the variance of WiFi signal strengths at various positions, as plotted in Fig. 14. The variance is smaller when the strength is sufficiently large or small and larger otherwise. Examining the plotted distribution, we assumed this relationship is approximated by a summation of a Beta function and a linear function of signal strength and fit it to the data to obtain the following function:

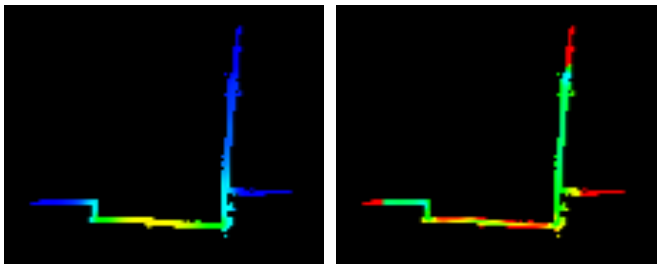
$$\begin{aligned} F_s(x) &= \frac{x^{4.59721}(100 - x)^{3.30446}}{2.9568824 \cdot 10^{13}} \cdot 8.32773 \\ &- 0.00388x + 5.18341. \quad (12) \end{aligned}$$

B. Person localization using particle filter

Predicted signal strengths are used for a particle filter-based person localization. The weight w_i of the ith particle, whose pose is $\boldsymbol{p}_i$, is given by:

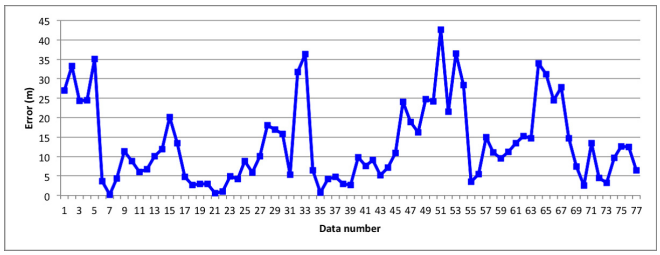
$$w_i = \sum_{n=1}^{N} LH_n \quad (13)$$

$$LH_n = \begin{cases} \frac{1}{\sqrt{2\pi\sigma_{in}^2}} \exp\left\{-\frac{(s_{in} - \mu_{in})^2}{2\sigma_{in}^2}\right\} & (s_{\text{in}} \neq 0) \\ 0 & (s_{\text{in}} = 0) \end{cases} \quad (14)$$

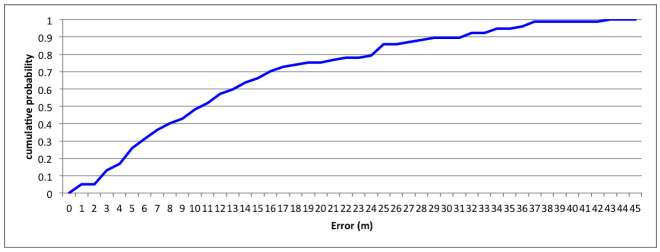


(a) Mean map. (b) Variance map.

Fig. 15. Pre-calculated WiFi signal maps.



(a) Error at each location.



(b) Cumulative error probability.

Fig. 16. Evaluation of WiFi-based localization.

where LH_n is the similarity for the nth AP (access point), which is set to zero when the signal for the AP is not detected. The weights are normalized over the all particles.

To reduce the calculation cost of the mean and the variance for every particle, we make their pre-calculated grid maps. Fig. 15 shows heat maps representing the magnitude of the mean and the variance for some AP. Using the map, the weight calculation can be done in real-time. For the prediction step of the particle filter, we use the number of steps estimated by a smartphone multiplied by an average movement per step of an ordinary person as a prediction of movement.

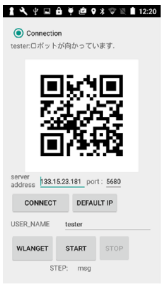
We tested the person localization method in a route in our building. The route is the left-side loop in the map shown in Fig. 8, where we use the mobile robot and collect WiFi signals with LRF data. Supposing that the localization results using the LRF data are correct, we calculate the error of the WiFi-based localization. Fig. 16(a) and (b) shows the point-wise errors and the accumulated error probability distribution, respectively.

The average error is about $13m$. Although this is worse than [12], this method recorded the averaged signal strength at each location with 60-second signal measurements, while our method uses a sequence of instantaneous measurements

Main GUI.

#	Description
①	IP Connect status
②	User name
③	Sensor measurements
④	Buttons for start and call
⑤	WiFi and GPS status
⑥	List of destinations

Description of each part of GUI.



Example QR code.

Fig. 17. GUI for smartphone application.

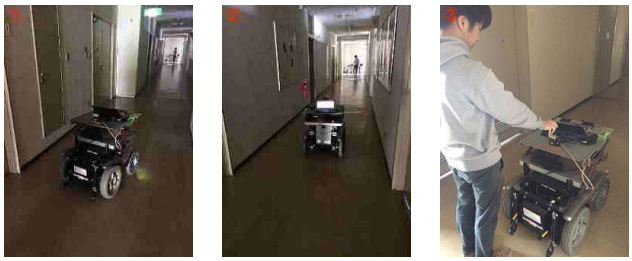
as the robot moves, and the time for making the WiFi signal maps is about ten minutes for the route shown in Fig. 16(a).

C. A robot call system

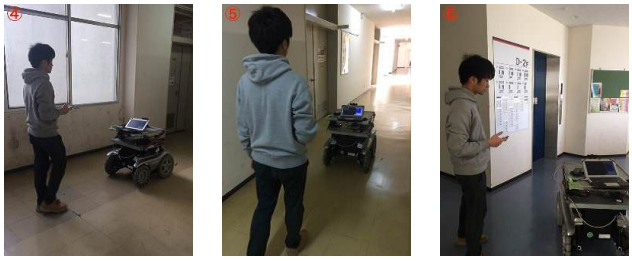
1) Overview of the system: Using the person localization method, we developed a prototype of robot call system, by which the user can call a robot and ask it to guide to some destination. The system is composed of a server system for receiving a call request and for dispatching a robot, an application on a smartphone of the user, and a robot control system on the robot. The application on a smartphone continuously sends WiFi signal data for updating the user's location on the server. When the user makes a call, the call request with the user's latest location is sent to the robot. The robot control system plans a route to the user and moves the robot. When the robot reaches the user, the user identifies himself/herself by showing a QR code, sent in advance from the server system, to the robot. Then the user specifies the destination from a list and the robot will guides the user to it. What the user has to do are: (1) push a call button, (2) show the QR code, and (3) select the destination. Fig. 17 shows the GUI of the application on smartphone. Fig. 18 shows snapshots of a successful use of the robot call system.

2) Reaching the user: The robot moves to the user autonomously based on the user location information sent from the server. The user localization is done by a particle filter, as explained above, and setting the destination of the robot to, for example, the center of mass of the particle distribution may not be appropriate when the uncertainty of user location is large. We thus apply a k-means clustering (k is currently set to twelve) to the particle set to extract a finite set of user location candidates. The robot takes a shortest path to visit all the candidates and stops when it finds a person.

3) Evaluation: We did a repetitive robot calling experiment. Numbers in Fig. 19. show the sequence of locations where the robot was called. We succeeded in calling all locations but the 4th location, where the scarcity of APs makes the user localization relatively less accurate and the robot failed to find the user. However, after giving up finding the user, the robot was able to continue the operation.



(a) The robot is called. (b) Moving to the user. (c) User authentication.



(d) Start guiding. (e) Continue guiding. (f) Guiding finished.

Fig. 18. An example use of the robot calling system.

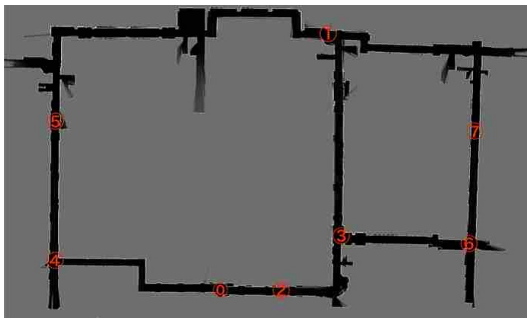


Fig. 19. Testing locations.

V. Conclusions and Future Work

This paper has described the use of WiFi signal data for improving both SLAM and person localization performance. We developed a WiFi-augmented loop closure detection for LIDAR scan-based 2D SLAM. By not making pose-to-pose correspondence directly but making sequence-to-sequence correspondence using WiFi signals, a reliable loop closure detection is realized. We successfully applied the method to shape feature-scarce indoor environments with large loops. The method is four times more efficient than a state-of-the-art image-based loop closure detection with a comparable detection accuracy.

We then developed a person localization method using a map with WiFi signal data. We model the distribution of WiFi signal strengths using Gaussian Process and the similarity between the measured and the predicted signal strength is used for calculating the likelihood. We implemented a particle filter-based person localization. We made a robot call system based on this person localization method and have shown that the person with a smartphone can be localized with a reasonable accuracy. Using the call system, the user can easily call a robot and make the robot guide him/her to some destination by just pushing buttons.

Currently the loop closure verification process using continuous DP matching runs as a batch process. One possible improvement is to separate the process as a backend one for a real-time mapping. Applying the proposed loop detection to 3D mapping of indoor environment with multiple floors is also an interesting extension.

The current WiFi-based person localization method uses signals from all APs in the environment. As we expand the area of consideration, the cost of making and using the WiFi signal maps also increases. One possible solution is to chooses only effective APs at each local area. Increasing the localization accuracy is also future work for reducing the exploratory movement of the robot near the person location. Evaluating the robot call system in a variety of situations for a long run is also necessary for future deployment.

References

[1] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.

[2] J.-L. Blanco, J.-A. Fernández-Madrigal, and J. Gonzalez. A new approach for large-scale localization and mapping: Hybrid metric-topological SLAM. In *Proceedings of 2007 IEEE Int. Conf. on Robotics and Automation*, pp. 2061–2067, 2007.

[3] M. Labbé and F. Michaud. Appearance-based loop closure detection for online large-scale and long-term operation. *IEEE Trans. on Robotics*, Vol. 29, No. 3, pp. 734–745, 2013.

[4] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal. Efficient, generalized indoor wifi graphslam. In *Proceedings 2011 IEEE Int. Conf. on Robotics and Automation*, pp. 1038–1043, 2011.

[5] P. Newman, D. Cole, and K. Ho. Outdoor SLAM using visual appearance and laser ranging. In *Proceedings 2006 IEEE Int. Conf. on Robotics and Automation*, pp. 1180–1187, 2006.

[6] M. Cummins and P. Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *Int. J. of Robotics Research*, Vol. 27, No. 6, pp. 647–665, 2008.

[7] T. Botterill, S. Mills, and R. Green. Bag-of-Words-driven, Single-Camera Simultaneous Localization and Mapping. *J. of Field Robotics*, Vol. 28, No. 2, pp. 204–226, 2011.

[8] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Proceedings of 2013 IEEE Int. Conf. on Computer Vision*, pp. 1470–1477, 2003.

[9] M. Labbé and F. Michaud. Online global loop closure detection for large-scale multi-session graph-based slam. In *Proceedings of 2014 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2661–2666, 2014.

[10] M. Labbé and F. Michaud. Memory management for real-time appearance-based loop closure detection. In *Proceedings of 2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1271–1276, 2011.

[11] B. Ferris, D. Fox, and N.D. Lawrence. WiFi-SLAM Using Gaussian Process Latent Variable Models. In *20th Int. J. Conf. on Artificial Intelligence*, pp. 2480–2485, 2007.

[12] S. Ito. WiFi Localization Method Using Gaussian Process Particle Filter. In *DICOMO 2011 Symp.*, pp. 439–448, 2011. (in Japanese).

[13] S. Segal, D. Hähnel, and S. Thrun. Generalized-ICP. In *Robotics: Science and Systems*, 2009.

[14] S. Thrun and M. Montemerlo. The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. *Int. J. of Robotics Research*, Vol. 25, No. 5-6, pp. 403–429, 2006.

[15] MRPT. `http://www.mrpt.org/`.

[16] NativeWifi. `https://msdn.microsoft.com/en-us/library/windows/desktop/ms706556(v=vs.85).aspx`.

[17] T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, and J. Sievänen. A Probabilistic Approach to WLAN User Location Estimation. *Int. J. of Wireless Information Networks*, Vol. 9, No. 3, pp. 155–164, 2002.

[18] R. Oka. Spotting Method for Classification of Real World Data. *The Computer Journal*, Vol. 41, No. 8, pp. 559–565, 1998.

[19] RTAB-Map. `http://introlab.github.io/rtabmap/`.