Machine Learning Approach to Self-Localization of Mobile Robots using RFID Tag

Yosuke Senta* Yoshihiko Kimuro* Syuhei Takarabe† Tsutomu Hasegawa†
*Institute of Systems & Information Technologies,, Fukuoka, Japan
†Kyushu University, Fukuoka, Japan

***Abstract*— This paper proposes a method for the self-localization of a mobile robot using a passive Radio Frequency Identification (RFID) system and Support Vector Machines (SVMs). Using the SVM, we do not need to perform any complicated tasks for measuring the geometric position of each RFID tags to produce a look-up table as used by conventional self-localization methods. Moreover, the method works even when several malfunctioning tags are included. The performance and accuracy of the method are confirmed by our simulation test, and we conclude that the method shows almost the same performance as that of a look-up table.**

***Index Terms*— Self-localization, Mobile robot, RFID, Support Vector Machine**

I. Introduction

A Radio Frequency Identification (RFID) System is an inexpensive device with high reliability. This device can attach an ID to each target material and read its ID without touching it. Many researchers have studied the accuracy of self-localization of a mobile robot using RFID tags as landmarks in the robot's environment [1, 2, 3]. In some of these studies, the RFID system is used with other sensors, such as cameras or laser range sensors (LRSs). In the other studies, the researchers define the positions of all the tags within the robot's environment as already-known information for the robot, and also place a large number of RFID tags in the environment to achieve higher levels of accuracy. However, it is infeasible, cost-wise, to set many RFID tags within the environment just for a mobile robot. On the other hand, it is true that the RFID tag is a very useful system. Therefore, it is expected that not only robot managers, but also other managers in different industries, will place RFID tags within their own environment for various purpose (e.g., geographical guidance, assistance for the handicapped, logistics management, etc.) sometime in the near future. If we can utilize existing RFID tags, previously placed within an environment by a third party, for the self-localization of a mobile robot, we would be able to achieve our goals of higher accuracy and lower costs.

Based on the above-mentioned idea, Yamano et al. [4] proposed a new method for the self-localization of a robot, through which the robot is supposed to 1) observe large number of RFID tags with unidentified location information, 2) obtain information about its location using a Support Vector Machine (SVM), and 3) localize its position. In this method, however, the researchers use active RFID tags, which release a radio wave from their own built-in battery. However, this method has not addressed the problems specific to passive RFID tags, such as the reception range, duration of power supply (charge time), and collisions between multiple tags.

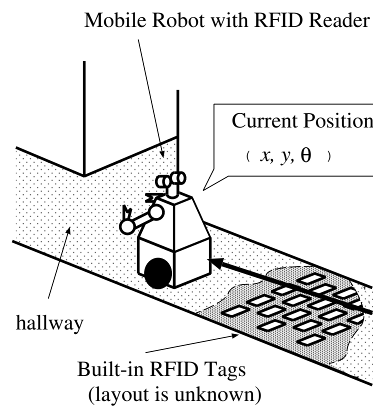

Fig. 1. The goal of this study is to establish a new self-localization system using passive RFID tags

Moreover, an active RFID tag has a troublesome maintenance problem because the battery needs to be replaced frequently. It is expected that active RFID tags, which are fixed in an environment, will soon be replaced with passive RFID tags. Therefore, we propose a new method of self-localization for a mobile robot, which addresses the above-mentioned problems, using a passive RFID system and machine learning.

II. Goal & Background

A. Goal

This study aims to design a self-localization method for a mobile robot using passive RFID tags placed on the floor, as shown in Fig. 1. Using this method, the mobile robot does not need to stop temporarily to observe the RFID tags. Moreover, this method does not require any information regarding the RFID tags except their ID. Considering the likely probability that RFID tags may have been placed within the robot's environment for different purposes, we employ the following scenario:

Step.1: Install RFID Tag Reader in a mobile robot, such as a cleaning or a monitoring robot, which accurately moves within the specified environment using some type of area sensors and/or dead reckoning.
Step.2: The robot collects the tag ID and learns the relationship between the tag ID and its own location information.
Step.3: The robot then transfers the result of the learning to

1-4244-1264-1/07/$25.00

other lower-functioned robot(s) in which only an RFID Tag Reader is installed.

Step.4: The lower-functioned robot(s) localizes itself (themselves) through the learning result and the ID of the RFID tags that have been observed.

The number of high-accuracy mobile robots used in this scenario is very small. Therefore, we do not have to use expensive range sensors and can establish the whole robot system with lower costs, which is one of the main advantages. Moreover, we are able to lower the costs of tag-layout, which is another advantage, because we do not have to measure the position of every tag on the floor.

B. Support Vector Machine

In this study, we used an SVM for robotic learning and tag ID in the same way as Yamano. The SVM has excellent general usability and robustness, and has been used in many fields recently including Text Categorization, Image Classification, Handwritten Digit Recognition, Bioinformatics, and others [5]. SVM is 2-class categorizer. To measure the location using a 2-class categorizer, we have to place SVMs on the points to be measured. This problem will be discussed in Section III-C. The SVM employs some types of calculation, called "kernel" or "kernel trick" to measure distance between two classes. There are two questions to be addressed before classifying events using SVM: 1) How should we vectorize an event? and 2) How should we decide the type of kernel to use and its parameters? Unfortunately, there are no decisive laws to determine these, and therefore we need to determine them heuristically. In this study, we used TinySVM [6]. The following kernels are supported in TinySVM: Linear ($\boldsymbol{W} \times \boldsymbol{X}$), Polynomial ($(s\boldsymbol{W} \times \boldsymbol{X} + r)^d$), neural ($\tanh(s\boldsymbol{W} \times \boldsymbol{X} + r)$), RBF ($\exp(s||\boldsymbol{W} \times \boldsymbol{X}||^2)$) and ANOVA ($(\sum_i \exp(s||W_i \times X_i||^2))^d$), where $\boldsymbol{X}$ is a vectorized event and $\boldsymbol{W}$ is the learning data.

III. Localization Method

A. Problem Setting

Our method does not localize a robot at a specified point within the environment, but estimates the distance between the robot and a wall at the time the robot passes through the specified area (Fig. 1). The passing moment can be judged from change in the color of the floor, the angle of the floor, etc. Therefore, when the robot passes through the hallway, the direction being measured is square to the hallway. At the specified area, it assumes that the location of each RFID tag is unknown.

There is an area of communication between an RFID tag and a reader in a RFID tag system. The extent of the communication area causes an error in the localization strategy. One simple and powerful approach to reduce this error is to use the location information gained by observing some RFID tags and then calculate the travel direction and location of the robot using the Least Square Method (LSM). We also examine LSM as a conventional method to compare it with our proposed strategy.

B. Vectorize

To utilize the SVM in machine learning, we have to vectorize the events that the robot needs to learn. These events include 1) ID of the tags read by the robot, 2) the ID-read count, and 3) the order of reading the tags. We carried out two types of vectorization using necessary information.

One of the two types of vectorization is carried out as follows: we indexed the ID of every tag placed within the environment as 0,1,2... and applied the tag ID-read count into each vector element

$$\boldsymbol{X} = (\frac{n_0}{n_{max}}, \frac{n_1}{n_{max}}, \frac{n_2}{n_{max}}, ...)^T. \tag{1}$$

Here, n_0, n_1, n_2... stand for the read count of each ID tag by the robot when it moves in the specified area, and n_{max} stands for the maximum values of the read count.

Another type of vectorization is as follows. First, we tried to reduce the dimensions of a vector. The former type requires the preparation of the same number of vectors as that of the RFID tags existing, within the environment. Therefore, if the number of RFID tags in the environment increases greatly, the system will be at risk of breakdown. Furthermore, the former type of vectorization does not consider the order of reading the ID tags. Based on these facts, we define the following formula:

$$\boldsymbol{X} = (k_0, \frac{n_{k_0}}{n_{max}}, k_1, \frac{n_{k_1}}{n_{max}}, k_2, \frac{n_{k_2}}{n_{max}}, ...)^T. \tag{2}$$

As shown in this formula, the indexed tag IDs read first, second, third... are described as k_0, k_1, k_2... respectively. In this type of vectorization, the dimension of a vector is equal to the two-fold number of ID tags that have been read, and therefore the dimension of a vector can be reduced.

In this paper, the first type of vectorization is referred to as Large Vectorize (LV) method and the latter as the Tiny Vectorize (TV) method.

C. Localization using SVM

Some improvements are needed to estimate the location of a robot when using the SVM because it is a 2-class categorizer, therefore, the single SVM can only judge whether a robot exists at a certain point. In this section, we address two types of SVM layout.

The first is called the "Parallel Method," which is similar to the method used by Yamamo. We placed SVMs along the measuring direction and checked which ones among them reacted to the measurement. Here, each SVM makes its own judgment, and therefore there is a possibility that several of the SVMs react to the measurement at the same time. To handle such a case, we decided to use an average of the data from these SVMs. In this method, the number of SVMs used is small, which is an advantage, but we have to make our calculation using all the SVMs.

The second method is called the "Tree Method." We piled up several layers of SVMs as a tree structure. In this system, an SVM in the first layer determines which part (right or left) of an event should be measured. When a part is measured, an SVM in the next layer under the measured part calculates. In

1-4244-1264-1/07/$25.00 ©2007 IEEE

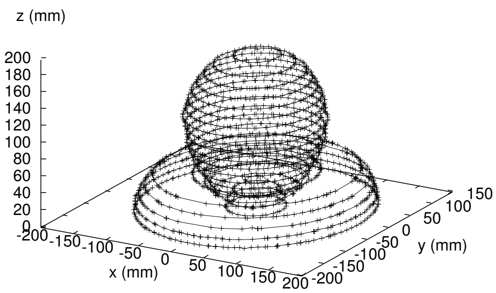

Fig. 2. Readable range of RFID System. The tag was set on $(0, 0, 0)^T$

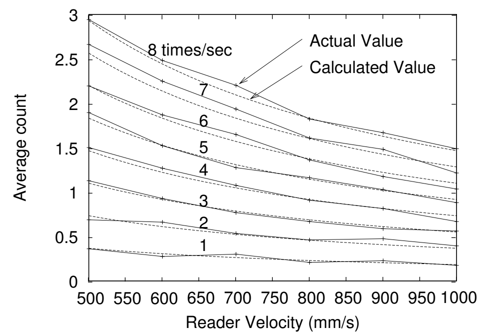

Fig. 3. Average read count between reader velocities. The actual value is similar to the value calculated by (3)

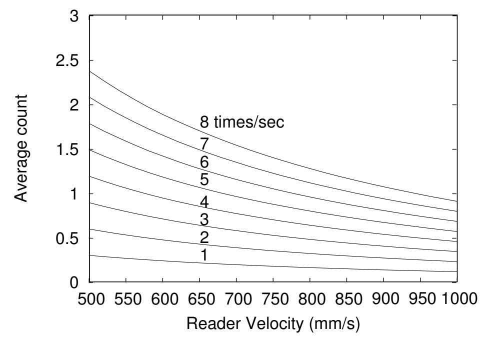

Fig. 4. Theoretical value of average read count in case of reading period $T = 70\mathrm{ms}$

this method, we use a large number of SVMs in total, but the number of actual working SVMs is small at any point in time. On the other hand, any misjudgment by an early layer SVM greatly affects the accuracy of the whole system.

IV. RFID SYSTEM

This chapter describes the properties of a passive RFID system, particularly the reading timing and the reading range of an RFID tag reader, both of which are strongly responsible for the self-localization of a mobile robot.

Fig. 2 shows the working range of the passive RFID system used in this study. The system used is TI's Evaluation Kit (RI-K3A-001A) and its carrier frequency is $134.2\mathrm{kHz}$. To obtain practical data, such as that shown in Fig. 2, we set a card-shaped tag at the origin $(0, 0, 0)^T$, and moved the tag reader in x, y, and z directions, respectively, for measurement using an arm robot. The card and the reader were set in parallel to the xy-surface. This figure shows that the working range is spread over $x = y = 0$ concentrically. In addition, the figure shows that if an RFID reader is set at a height of $100\mathrm{mm}$ from the floor, the working range of a tag is a circle with a diameter of $183\mathrm{mm}$. This value is used in an experiment described in the next chapter.

On the other hand, a mobile robot reads every tag it moves over. Therefore, we need to measure how the relative speed between a tag and a tag reader would affect the readability. First, we fixed n, which stands for count of tag reading per second, then moved the tag reader between $(-225, 0, 100)^T \sim (225, 0, 100)^T$ in a linear motion with uniform velocity, and measured how many times the tag reader was able to read the tags. Fig. 3 shows the relationship between the average tag-read count from 150 trials and the velocity of the tag reader. The "calculated value" marked in the figure stands for a linear line calculated using the following formula: the working range $L = 183\mathrm{mm}$ was divided by the distance a tag reader moves when it reads a tag at one time.

$$m = \frac{Ln}{v}. \tag{3}$$

Here, v stands for the velocity of the tag reader.

A passive RFID tag requires a certain period of time for power supply and communication. In the case of the device used in this study, the duration periods for the power supply and communication were $50\mathrm{ms}$ and $20\mathrm{ms}$, respectively; the duration time was $70\mathrm{ms}$ in total. Although, the values calculated by (3) were similar to the measured values, we did not consider this duration period. For the case where the tag reader was outside (inside) the working range and came into (got out of) the working range within $70\mathrm{ms}$, the tag reader failed to read the tags; therefore, the average tag-read count was lower. Fig. 4 shows the theoretical values for the case where the time needed for reading the tags was $70\mathrm{ms}$. As shown in the figure, the theoretical values are different from the actual measured values. This is probably because the chargeable range is larger than the communicatable range. For the case of the device used in this study, we found that if the tag reader was in the working range, the tag reader could always read the data.

V. SIMULATION

A. Simulation System

As for machine-learning, we have to change the combination of tag-layout and experimental parameters diversely for a true evaluation, and therefore it takes a long time to test all the combinations using an actual model. Therefore, we established a simulation system based on actually measured data of the RFID system mentioned in Chapter IV. In this simulation system, we assumed a hallway with a width of x, and laid several RFID tags in a y-long-square area (Fig. 5). The layout of the tags could be changed freely. The tag-reading interval $1/n$ and the working range L of the tag reader were also determined freely. The moving speed, v and the moving direction of a mobile robot could also be changed freely. Since

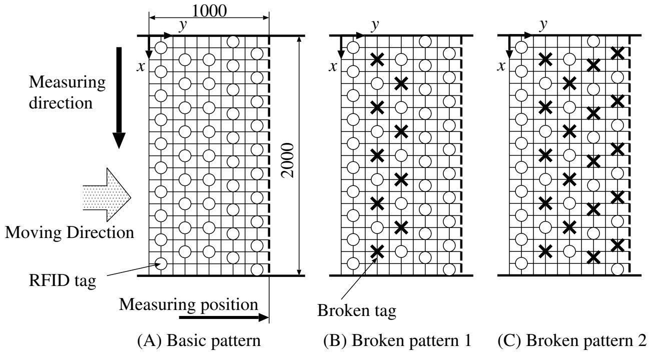

Fig. 5. Tag layout patterns for several simulations

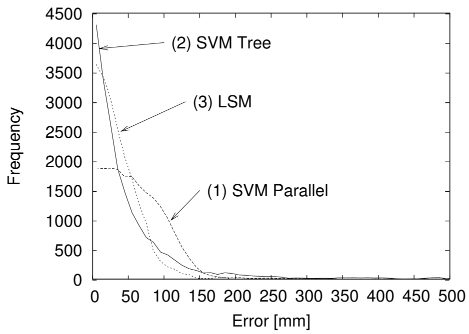

Fig. 6. Simulation result between SVM layouts. This used Linear kernel and LV-Method, 600 teaching data

the tag layout was defined as unknown for the robot, the starting and terminating points or the movement of the robot were selected at random. In addition, the starting time for the tag reader to read the tags was selected at random between 0 and $1/n$.

This simulation test was conducted in three different phases. In the first phase, we obtained a set of status vectors, which is supposed to work as training data obtained by many times of robot passage simulation. In the second phase, we obtained teaching data by using the training data. In the final phase, we obtained a status vector and assumed the location of the mobile robot by the SVM.

We used the following conditions for this simulation test: $x = 2000\text{mm}$, $y = 1000\text{mm}$, $1/n = 100\text{ms}$ ($n = 10\text{Hz}$), $L = 183\text{mm}$, and $v = 1.1 \sim 5\text{m/s}$. Fig. 6 shows the comparison between the Parallel Method and Tree Method, which were mentioned in Section III-C, under the tag pattern Fig.5 (A). For this graph, we conducted 20,000 robot passage simulations and then calculated the absolute values between the actual values and the estimated ones every 10mm. We placed 64 SVMs along the measuring direction for the Parallel Method and made a pile of six layers for the Tree Method. We employed the simplest combination of LV & Linear Method with 600 as the number of teaching data for the status vector and kernel trick, since this simulation focused on the comparison between the two SVM methods. This figure shows that the Tree Method has almost the same capability of assuming the location of a robot as the LSM. On the other hand, the Parallel Method has a poor performance since several SVMs in the system react to the measurement at the same time. As for the Parallel Method, we do not mention it again because of space limitation.

TABLE I

LOCALIZATION ABILITY OF DIFFERENT KERNELS

Kernel Trick	LV-Method	TV-Method
Linear	4756	657
Polynomial	7671	-
Neural	-	-
RBF	7225	4810
ANOVA	1475	795

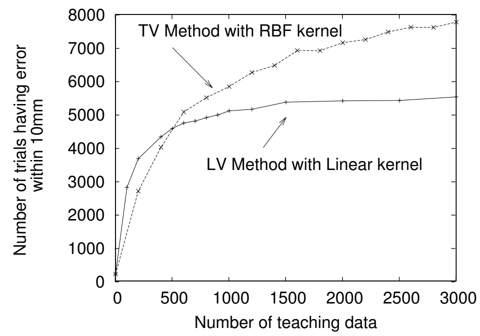

Fig. 7. Relationship between measurement accuracy and number of teaching data

B. Kernel Trick and Vectorizing

Next, we conducted a comparison to determine which combination was the most appropriate for the kernel trick, and the vectorization methods between LV and TV. We employed the Tree Method for the SVM layout and set 600 as the number of Teaching Data. We set s, r, and d, which are the parameters in the kernel trick, as 2, 1, and 2 respectively. Table I shows thes results. Each value in the table shows how many times the allowance ended in less than $\pm 10\text{mm}$ with 20,000 simulations. The higher the values, the better the performance of the system. The hyphenated columns show that we failed to record the measurement since the learning failed to converge. Even when we changed the parameters of the kernel trick, the learning failed to converge or the end result was about the same.

Table I shows that Linear and Polynomial kernels are desirable in LV, and RBF is desirable in both methods. We also decided, in terms of vector-saving, to continue our research mainly on the TV & RBF Method (TVRM) and on the LV & Linear Method (LVLM), because this is the simplest method as a control target for comparison with TVRM.

C. Comparison by Size of Teaching Data

The number of teaching data is related to the accuracy of results. Fig. 7 shows the relationship between the number of teaching data and the measurement allowance in the above-mentioned combinations, which was less than 10mm. The figure shows that as the number of teaching data increases, the accuracy improves, but the improvement remains constant. We can point out that there is a trade-off relationship between

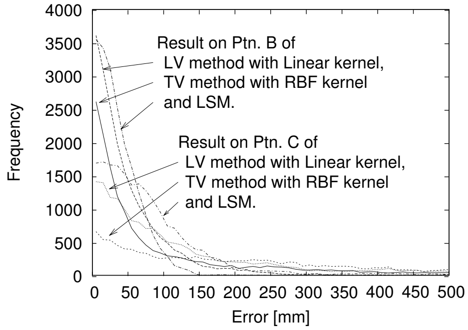

Fig. 8. Simulation result on decreasing tags. (Teaching data: Fig. 5 (A), 600 of teaching data for LV-Linear Method, 1,800 for TV-RBF Method)

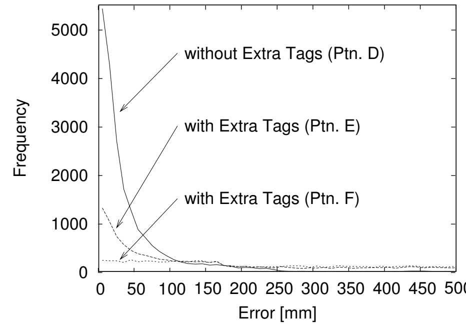

Fig. 10. Simulation result on increasing unknown tags. (Teaching data: Fig. 9 (F), TV-RBF Method, 1,800 of teaching data)

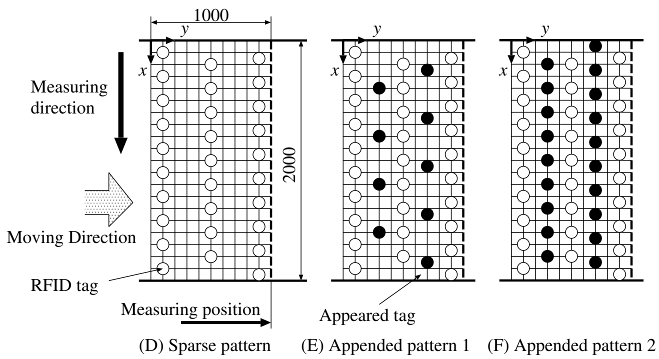

Fig. 9. Tag layout patterns for simulation with increasing RFID tags

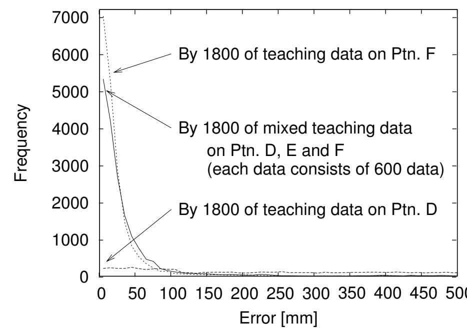

Fig. 11. Simulation result on increasing tags by mixed teaching data. Current tag pattern is Fig. 9 (F). (TV-RBF Method, 1,800 of teaching data)

calculation loads due to the increase in the number of teaching data and accuracy thus obtained. Therefore, it is not desirable to increase the number of teaching data without having some form of limitation. Therefore, we decided to use 600 and 1,800 of LVLM and TVRM, respectively in the following research. These values show the number of teaching data with which we are able to obtain 90% accuracy of the highest limit.

D. Tolerance of Increase or Decrease in Tag Amount

RFID tags usually remain in an environment for a long time, and suffer deterioration caused by different types of outer load, including the weight of people who walk on them, or heavy cargos that are transported across them. In this case, the number of RFID tags in the environment decreases because the damaged tags make virtually no response. On the other hand, third parties sometimes lay new RFID tags in an environment and thus the number of RFID tags increases. We now study the relationship between the increase or decrease in tag amount and the capability of a robot's self-localization.

First, we studied the case of a decrease in RFID tags. We used the tag layout shown in Fig. 5 (A) as the learning data and considered a case where the tags were damaged, as shown in Fig. 5 (B) and (C). Fig. 8 shows the results. Note that the values in TVRM almost reach 7,000 at its peak and when it does not break down, as shown in Fig. 7, Fig. 8 shows that TVRM is very sensitive to the breakdown of a tag since the tags are placed in a small area with the sensor side up.

Next, we moved on to the case with an increase of RFID tags. We used the tag layout shown in Fig. 9 (D) as the learning data and considered a case where the tags increased in number, as shown in Fig 9 (E) and (F). In LVLM, new tags are regarded as separate new elements of different dimensions and as having no relation to the learning strategy. Therefore, the performance did not deteriorate. On the other hand, with TVRM, the above-mentioned scenario does not occur. Therefore, in the case of TVRM, where there is an increase in the number of new tags, robot's self-localization capability deteriorates as shown in Fig. 10. However, this disadvantage can be avoided by the use of the method of being ignored tags which does not appear in the learning phase.

In addition, this algorithm requires hundreds of learning data. It is far from an efficient operation if learning data from older generations is deleted every time a new tag is placed in the environment. Therefore, we estimated what robot's self-localization capability would be if we used learning data from mixed generations. Fig. 11 shows the simulation results. In this test, learning data from different generations of Fig. 9 (D), (E), and (F) were uniformly mixed and used to self-localize the robot. Fig. 11 shows the effectiveness of being able to delete and replace the learning data from older generations with new data.

VI. Experiment Using Wheelchair Robot

We conducted a demonstration experiment using a wheelchair robot to confirm the effectiveness of the above-mentioned algorithm. Fig. 12 shows the appearance of the wheelchair robot. This robot has an LRS on both sides, which can measure the distance of an obstacle. The robot also has an

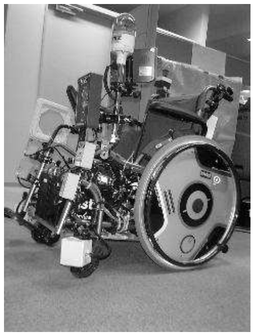

Fig. 12. Photo of Wheelchair robot

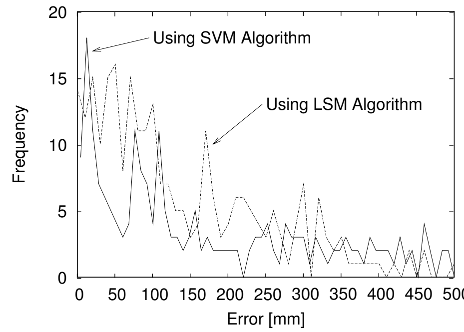

Fig. 14. Experiment result using wheelchair robot

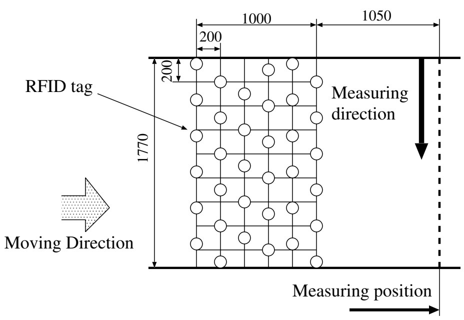

Fig. 13. Tag layout pattern for the experiment

RFID tag reader, which was mentioned in Chapter IV, placed centrally at the bottom of its main body so that it can read the RFID tags placed on the floor.

In this experiment, the robot was moved around the tag-placed floor as shown in Fig. 13. The robot can measure the actual distance from the wall by the LRS. We compared the measured and estimated values from the RFID tags and SVMs. In this experiment, we obtained only 280 pcs of data in total; therefore, we could not obtain the learning data using of the actually-measured values. Therefore, we first prepared 1,800 pcs of teaching data using the simulator mentioned in the previous chapter, and then estimated the location of the robot using this data. Fig. 14 shows the experimental results. As for the SVM in this figure, we employed the TV-RBF Method. This figure clearly shows that this method has almost the same performance as the conventional method of LSM when using the hypothetical teaching data provided by the simulator. However, the measurement error is large, on the whole, including that of the LSM. We expect that this is caused by the noise of LRS, errors in tag layout, etc. Even in case where the errors in tag layout are mainly responsible for the measurement error, its possible adverse influence will disappear if we use the measurement data as the learning data. This assumption should be confirmed, which is an important task for the future.

VII. Conclusions

We have proposed a new method for the self-localization of a mobile robot that is supported by passive RFID tags and a Support Vector Machine (SVM). This new method improve self-localization by enabling a robot to localize itself, even if it does not know the location of the tags placed in its environment. Therefore, we are able to use a great number of RFID tags for this purpose.

From simulations of a mobile robot with the RFID system, we determined the most appropriate SVM layout, the most efficient vectorization and kernel trick, the most practical number of teaching data, etc. We also applied this algorithm to an actual wheelchair robot, and found that this new method has approximately the same performance as the Least Squares Method.

Acknowledgments

We would like to thank Dr. K. Murakami of Kyushu University for his help in the production of a robot arm specifically designed for evaluating RFID properties. Our thanks are also extended to Shimizu Construction Co.,Ltd. for lending us a wheelchair robot. This research was financially supported by the 2005 Scientific Technology Promotional Adjustment Budget, provided by Ministry of Education, Culture, Sports, Science and Technology in Japan.

References

[1] V.Kulyukin, C.Gharpure, J.Nicholson and S.Pavithran, “RFID in Robot-Assisted Indoor Navigation for the Visually Impaired”, *Proc. of the 2004 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2004

[2] D. Hähnel, W. Burgard, D. Fox, K.Fishkin and M.Philipose, “Mapping and Localization with RFID Technology”, *IEEE Int. Conf. on Robotics and Automation*, pp.1015-1020, 2004

[3] T. Tsukiyama, “Navigation System for Mobile Robots using RFID Tags”, *Proc. of ICAR 2003*, pp.1130–1135, 2003

[4] K. Yamano, K. Tanaka, M. Hirayama, E. Kondo, Y. Kimuro and M. Matsumoto, “Self-localization of Mobile Robots with RFID System by using Support Vector Machine”, *Proc. of 2004 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.3756–3761, 2004

[5] N. Cristianini and J.Shawe-Taylor, “An Introduction to Support Vector Machines”, *Translated from Cambridge University Press 2000 by T. Ookita*, Kyoritsu Publisher, 2005 (In Japanese)

[6] `http://www.chasen.org/~taku/software/TinySVM/`