SELF-LOCALIZATION SYSTEM FOR ROBOTS USING RANDOM DOT FLOOR PATTERNS

*Yutaro Fukase[1] and Hiroshi Kanamori[1], Shinich Kimura[2]

[1]*Shimizu Corporation*
3-4-17, Etchujima Koto-ku
Tokyo, Japan, 135-8530
*(*Corresponding author: fukase@shimz.co.jp)*

[2]*Tokyo University of Science*
2641 Yamazaki, Noda-shi
Chiba-ken, Japan, 278-8510

SELF-LOCALIZATION SYSTEM FOR ROBOTS USING RANDOM DOT FLOOR PATTERNS

ABSTRACT

Various types of service robots have recently been developed for guarding facilities, caring for the elderly, carrying objects, and cleaning buildings. As barrier-free facilities improve and their use expands, these robots have more space within which to move inside buildings. Yet robots that move autonomously rely on position-detection systems. Though improving rapidly, these systems are far from perfect in determining positions in certain situations, especially when robots navigate large areas or cross various locations. Our group is working to solve this problem by developing a position-detection system using random-dot patterns on a floor. First, we construct a floor with a random-dot pattern and register the positions of all of the dots into a database. As the robot moves across the floor, a camera on the robot captures an image of the floor beneath it and crops the dot pattern in the image. The cropped dot pattern is matched to the dot patterns in the database to determine the position of the robot and the direction in which the robot is facing or moving. In this paper we propose a self-localization system and matching algorithms derived from a space technology and present the results of several experiments.

KEYWORDS

Self-localization system, Matching algorithm, Space technology

INTRODUCTION

A service robot programmed to move autonomously in a building must be equipped with a self-localization system to operate effectively. Several types of self-localization systems are therefore being pursued.

A number of strategies using electric devices installed inside buildings have been developed. One system deploys a pseudo satellite (Sakamoto, Niwa, Ebinuma, Fujii, & Sugano, 2010); another detects differences in radio field intensity among several wireless LAN access points (Umetani, Yamashita, & Tamura, 2011); a third uses an ultrasonic 3D position sensor (Nishida & Takeda, 2010). Yet in each case, non-uniformity of the electromagnetic or acoustic environment causes instability the detection position.

Other methods rely on ceilings or floors for position detection. One, for example, uses arrayed marks on ceilings (Nakazato, Kanbara, & Yokoya, 2008), while another tracks communications transmitted LEDs on ceilings (Uchiyama, Haruyama, & Nagamoto, 2009). Both of these methods fail when the ceilings in a room are tall. Kodaka, Niwa, and Sakamoto (2009) developed a floor-based system by placing a series of RFID tags on a floor and reading them to detect positions. At about the same time, Nishizaka, Hiyama, Tanikawa, and Hirose (2009) printed position-coded patterns on a floor and read them with a camera. These latter floor-based systems are costly, however, as the first requires many RFID tags and the second requires the printing of code patterns on a floor. Another floor-based system relies on cracks in a floor as landmarks (Kelly, 2000). This method is difficult to adopt for floors with finishing materials, though it is effective for factories or warehouses with solid concrete floors.

Our group is attempting to surmount these challenges by devising a method using random dot patterns on floors. Floors are often paved with vinyl or coated with epoxy resin in structures such as hospitals, factories, or office buildings. To improve the appearance and durability and to recycle materials, some floors are finished with dot patterns formed by mixing milled plastics into the flooring materials at the manufacturing or construction phase. These dot patterns are random, hence one pattern at one position is statistically different from all of the other patterns. Our group has taken advantage of this property to develop a practical self-localization system for mobile robots with a focus on low cost and high detection stability.

THE PROPOSED SYSTEM

Outline

The basic configuration of the proposed system is shown in Figure 1. After constructing a floor with a random dot pattern, we scan the floor and make a dot-position database. As the robot moves, a camera on the bottom captures images of the dot patterns. To determine the robot's position and direction of travel, the system matches the position and orientation of the dot pattern in the captured image with the data in the dot-position database of the entire floor.

The dot patterns in the image are matched with those in the dot-position database using an attitude-detection technology applied for satellites. In a satellite, an on-board camera captures images of the sky and the camera orientation is identified by matching the stellar constellation in the captured image with a known star map. This technology, otherwise known as the *Star sensor* technology, is illustrated schematically in Figure 2. Our detection method shares the following elements with the Star sensor method.

1. Naturally occurring random pattern

Both the star constellation and floor dot pattern are random dot patterns.

2. Foreign dots and disappearing dots

Countless stars may appear in a satellite image, hence the stars used for matching are limited to those above a preset threshold for brightness. Yet in the brightness range just above and below this threshold, non-target stars may inadvertently appear or target stars may disappear. The same principal applies in our system, as previously unregistered dots (hereafter, "foreign dots") may inadvertently appear or database-registered dots may disappear.

To cope with this problem we apply the *Polestar algorithm* (Silani & Lovera, 2006), an attitude-detection algorithm with a robust ability to handle the blending of non-target stars or the disappearance of target stars.

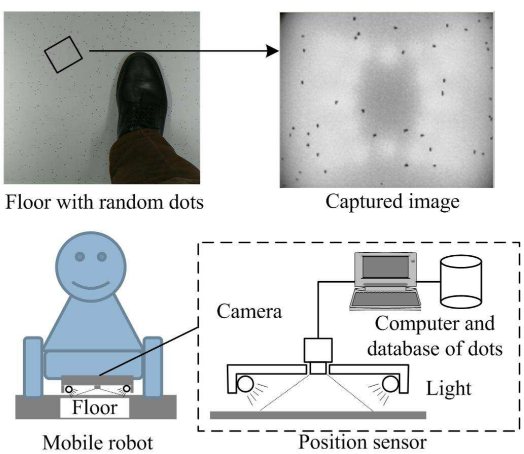


Figure 1- Proposed system

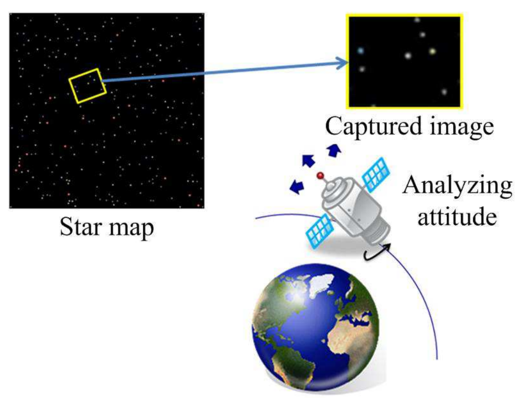


Figure 2 - Star sensor

Method

Basic Principle

In a random dot pattern, a combination of distances between a central dot and neighbouring dots within a certain distance is statistically different from another combination with another central dot, provided that the latter dot is sufficiently different. We refer to this combination of distances as the “polestar characteristic” of a dot. The Polestar algorithm uses this polestar characteristic to match a dot in the image with a dot in the database. We compare the polestar characteristic of a dot in the image with the polestar characteristic of each dot in the database and count the number of same distances. The matching dot in the database is the dot with the highest number of same distances.

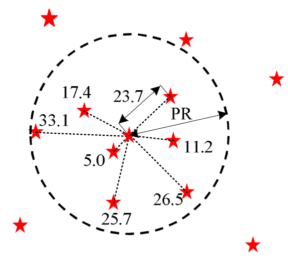


Figure 3- Polestar Characteristic

In Figure 3, for example, there are seven dots within a “pattern radius” (PR) from a central dot, and the polestar characteristic is [5.0, 25.7, 26.5, 11.2, 23.7, 17.4, 33.1]. The PR has to be adequately specified here. For our method, we define the PR as the diagonal length of the captured image.

Polestar Database Matrix

The “polestar database matrix” is a matrix of the polestar characteristics of all of the dot patterns on the floor. As an example, Figure 4 presents the process used to calculate the polestar characteristic of the i-th dot. At this point, every distance between the i-th dot and each outer dot is round up to the whole number, “a distance index.” The PR is round up to the whole number “PRidx.” We make a PRidx length bit vector, “PSi”, in which each bit corresponding to the distance index becomes 1 and the other bit becomes 0. Calculating the bit vectors associated with each dot in the database and arraying those vectors, we generate the polestar database matrix (**PSdb**). Even if more than one neighboring dot has the same distance index, the corresponding bit of the vector becomes 1. This procedure is executed once before an actual matching process.

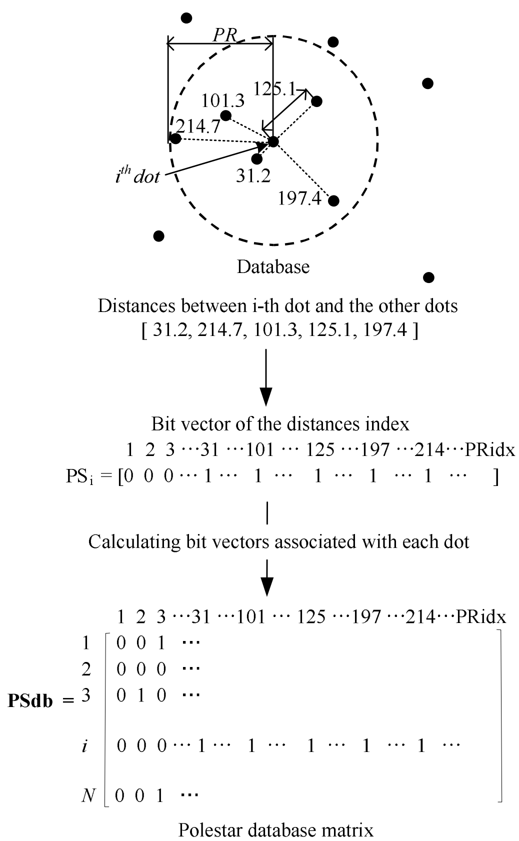


Figure 4 - Polestar database matrix

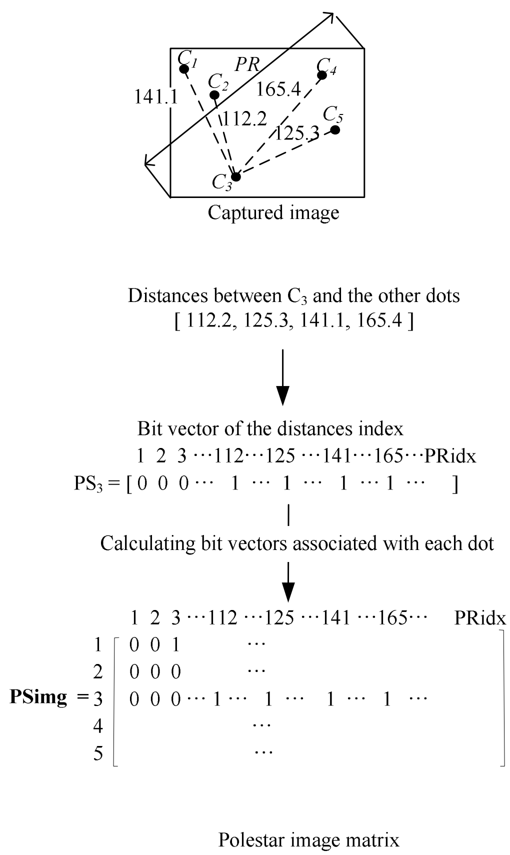


Figure 5 - Polestar image matrix

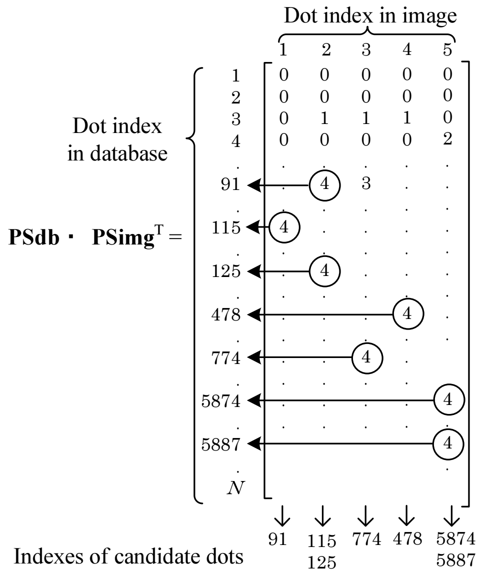


Figure 6 - Matching multiplication

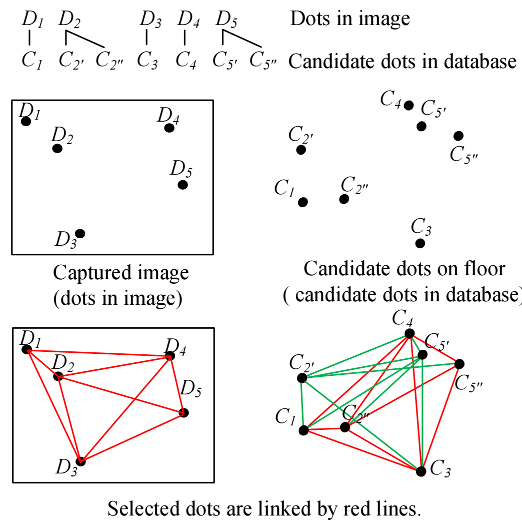


Figure 7 - Selecting combination

Polestar Image Matrix

The first step in the actual matching process is to make a "polestar image matrix" (**PSimg)** by the same method used to generate the pole star database matrix. Figure 5 presents a calculation process.

Polestar Matrix Multiplication

Figure 6 shows the multiplication of **PSdb** with the transposed **PSimg**. The number given for each matrix element of the calculated matrix is the number of matched polestar distances between each dot in the database and each dot in the image. To find the candidate dot in the database matching the i-th dot in the image, we only have to find the matrix element with the maximum number in the i-th column and the row index of this element, which corresponds to dot index in the database. If the maximum number exists at more than one matrix element in the i-th column, more than one candidate dots are selected.

Selecting an Adequate Combination

In the previous process, several dots in an image correspond to more than one candidate dots in the database. For example, in Figure7, D_2 corresponds to C_2' and C_2", and D_5 also corresponds to C_5' and C_5". If every dot in the image adequately matches the dot in the database, a distance between the each two dots in the image and the distance between the two corresponded dots in the database are of the same length. Using this relation, we can select a matched combination of dots in database. In Figure7, the length of the red lines between each dot in the image and those in the database are equal. And then with the positions of the selected dots in the database, we calculate the position and orientation of the image on the floor.

Matching Rate

In advance we virtually divide the floor into grids with a cell whose resolution roughly sets to the averages of the dot dimensions. And if a dot exists in the cell, the cell becomes "Existed Cell," and if not, it becomes "Empty Cell." Corresponding to the position and orientation calculated in the previous procedure, every dot in the image is translated into a floor coordinate system and is assigned to the Existed Cell or the Empty Cell. The number of dots in the Existed Cell divided by the number of all dots in the image is the “matching rate.” We describe the procedure in Figure 8. This matching rate is useful for evaluating the reliability of the matching and eliminating positions detected erroneously due to low matching rates.

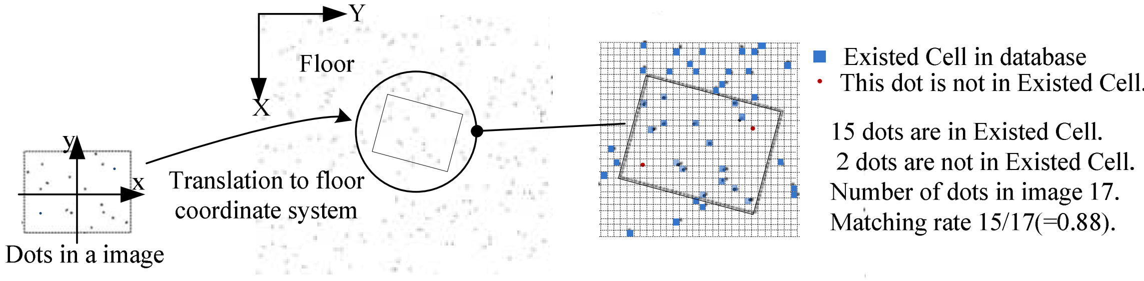


Figure 8 - Calculating matching rate

PERFORMANCE VALIDATION

To examine the effects of newly appearing, foreign dots (dots not previously detected or stored in the database) and disappearing dots (vice versa) we conducted an experiment with actual flooring and a computer simulation experiment with a dot database of the actual floor.

Experimental Configuration

The flooring material was a resin embedded with black plastic chips with diameters ranging from around 2 to 3 mm. 91,582 black dots were on 1800 mm x 9450 mm. The distribution density of the black plastic chips could not be strictly controlled, but the number of chips added could be adjusted to about 30 to 40 chips per field of view (FOV: 90 mm x 90 mm). A camera captured an image of the floor under white LED illumination. Then we conduct binarizing, denoising, and labelling, and calculate positions of the black dots in the image. Finally, the positions were corrected to compensate for lens distortion.

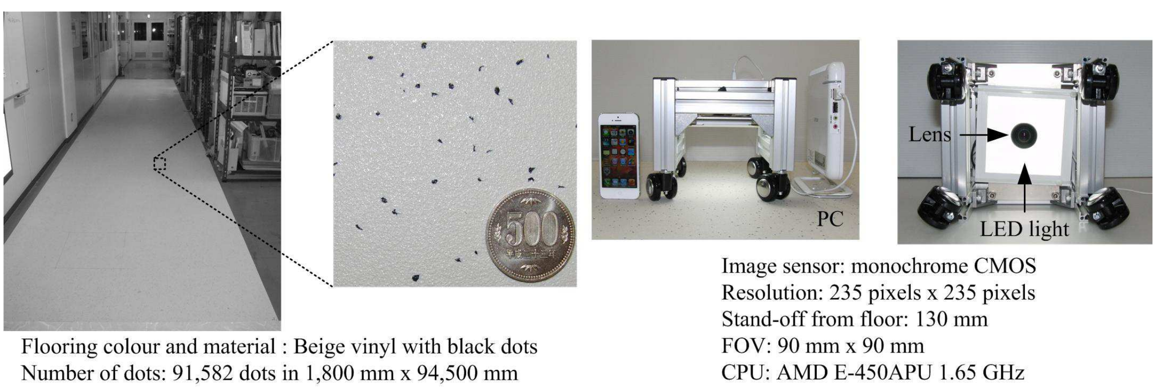
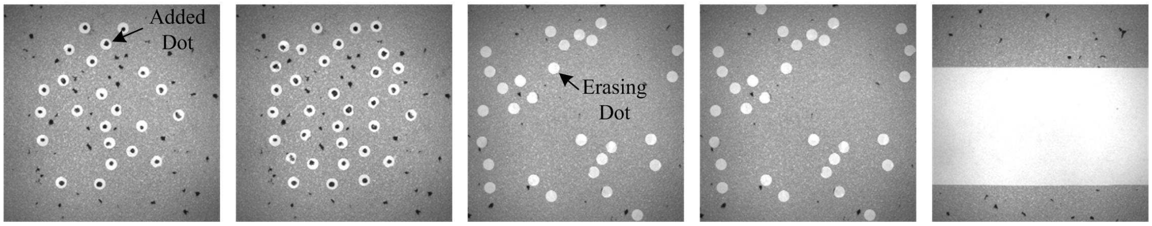


Figure 9 - Experimental configuration

Actual Experiment

After adding several foreign dots or removing several dots on the floor, we detected the position of the captured image. Figure 10 shows five cases conducted.

Both added dots and naturally generated dots were captured in the image. Tiny foreign particles and long and tiny hairs were eliminated by image-processing, and several small dots in the database tended to disappear. On the other hand, the database of dots on the floor was prepared in a clean environment on the assumption that no dots would be added. As a result, relatively small dots were used as reference dots on the floor and the tiny dots in the database were sometimes unrecognizable as dots in the image. Table 1 indicates the percentage of added dots, the percentage of removed dots, and the detection result for the respective cases. Detection was still successful when 61.0% of the dots were added (Case 1) or when 76.2% of the dots were removed (Case 3). The experiment confirmed a robust detection performance.



Case1 Adding 23 dots | Case2 Adding 33 dots | Case3 Erasing 28 dots | Case4 Erasing 30 dots | Case5 Erasing dots with white line

Figure 10 - Experimental cases

Table 1 - Experimental results

	Case1	Case2	Case3	Case4	Case5
Adding Rate(%)	61.1	90.2	9.5	9.5	7.8
Erasing Rate(%)	10	14.6	76.2	81	78.4
Detection	CD	ND	CD	ND	CD

CD, Correctly detected; ND, Not detected

Simulation Experiment

Sampling Dots from the Database

Next, we executed a series of simulation experiments for statistical consideration with many more sample cases. The conditions of the simulation conformed to the actual experiment. We simulated the capture of 1,209 images at 112.5 mm intervals on the actual floor. The dots captured at each point were taken from the database. To approximate the observational error, every dot was given a random position error within 1 pixel.

Adding Dots

We randomly added several dots to the sampling dots mentioned above. Table 2 shows the result of the detection position in the case with added dots. Correct positions were detected at every position with a rate of dot addition of 30 % or less. With a rate of 50 % added dots, 9 out of 1,209 were undetected or erroneously detected. Therefore, the position-detection performance appears to be robust to added dots.

Removing Dots

We simulated the disappearance of dots by removing points from the sampling dots mentioned above. Table 3 shows the result. Even in the case of a 50 % rate of dot removal, over 95 % of the positions were correctly detected. The position-detection performance is also robust to dot removal.

Adding and Removing Dots

Dots can naturally be expected to appear and disappear simultaneously on an actual floor. By analyzing captured images in various positions, we found that the rate of dot disappearance in an image was 25 % at most. Hence, we added dots at the several different rates and detected positions in an image with 25 % of the dots removed. As in Table 4, about 90 % of the positions were correctly detected in an image with dots added at a rate of 50 %.

Eliminating Erroneous Detections

Erroneous detection can lead to dangerous malfunctions in mobile robots. A lower value is set to the matching rate to avoid erroneous detection. In an examination of the matching rates in Tables 2 to 4, we found the lowest matching rate for the correct detection position and the highest matching rate for the erroneous detection position in Table 4. Table 5 shows the matching rates in Table 4. The lowest matching rate for correct detection was 38.9% and the highest matching rate for erroneous detection was 23.5%. Hence, when the standard matching rate is set between 23.5 and 38.9 %, the erroneous detection can be eliminated with no sacrifice of correct detection in every case. In addition, since the dots on the floor were around 3 mm in diameter, we set the resolution for existing dots to 2.8 mm.

CD, Correctly detected; WD, Wrongly detected; ND, Not detected

Table 2 - Adding dots

Adding(%)	10	20	30	40	50
CD(%)	100	100	100	99.59	99.26
WD(%)	0	0	0	0.33	0.66
ND(%)	0	0	0	0.08	0.08

Table 3 - Removing dots

Erasing(%)	10	20	30	40	50
CD(%)	100	99.92	99.59	99.5	95.95
WD(%)	0	0	0.17	0.08	0.91
ND(%)	0	0.08	0.29	0.4	3.14

Table 4 - 25 % dot removal and dot addition of several %

Adding(%)	10	20	30	40	50
CD(%)	99.57	98.76	97.35	95.54	89.66
WD(%)	0.08	0.5	1.41	3.39	7.44
ND(%)	0.33	0.74	1.24	2.07	2.9

Table 5 - Matching Rate

Adding(%)	10	20	30	40	50
Min. matching Rateof CD(%)	61.1	56.3	51.5	38.9	42.9
Max. matching Rate of NG(%)	10	16.1	23.5	18.9	21.4

SUMMARY

We proposed a method to detect positions with random dot patterns, conducted an experiment and simulation on actual vinyl flooring, and confirmed the following:

1. The method is robust to the addition of dots and the removal of dots.
2. Misdetection can be avoided by appropriately configuring a standard matching rate.

The floor tested in the actual and simulation experiments covered an area of 1800 mm x 9450 mm. With other simulation, however, we succeeded to detect positions on a virtual floor database covering an area ten or more times larger with pseudorandom dots. In that case, it takes 30 to 40 seconds to detect a position, so we have been trying to speed up the process with GPU.

In this paper we show an important part of our self-localization system for detecting initial position. Once initial position is detected, the searching area can focus on the small area around the detected position and amount of time for calculation of detecting next position can be decreased dynamically. And even if the detecting position fails, with detecting the movement of the dots between two successive captured images, we can calculate the position continuously. Presently we are applying this self-localization system to control a mobile robot.

REFERENCES

Yoshihiro Sakamoto, Haruhiko Niwa, Takuji Ebinuma, Kenjiro Fujii, Shigeki Sugano (2010). *Indoor Positioning with Pseudolites*. SICE Annual Conference 2010, CD-ROM

Tomohiro Umetani, Tomoya Yamashita and Yuichi Tamura (2011). *Probabilistic Localization of Mobile Wireless LAN Client in Multistory Building based on Sparse Bayesian Learning*. Journal of Robotics and Mechatronics (vol. 23, no. 4, pp. 475-483).

Yoshifumi Nishida, Hideki Takeda (2010). *Ultrasonic Three-Dimensional Location System*. Journal of the Society of Instrument and Control Engineers(vol. 49, no. 1, pp. 56-59).

Yusuke Nakazato, Masayuki Kanbara, Naokazu Yokoya (2008). *A User Localization and Marker Initialization System Using Invisible Markers*. Journal of the Virtual Reality Society of Japan (vol. 13, no. 2, pp. 257-266).

Hideaki Uchiyama, Shinichiro Haruyama, Naoki Nagamoto (2009). *Construction Machinery and Equipment* (vol. 45, no. 2, pp. 50-53).

Kenri Kodaka, Haruhiko Niwa , Yoshihiko Sakamoto (2009). *Robot Pose Estimation on Floor Equipped with a Lattice of RFID Tags*. Transactions of the Society of Instrument and Control Engineers (vol. 45, no. 7, pp. 379-387).

Shinya Nishizaka, Atsushi Hiyama, Tomohiro Tanikawa, Michitaka Hirose (2009). *Robustness Enhancement in Localization using Interior Decoration with Coded Pattern*. Technical Report of the Institute of Image Information and Television Engineers (vol. 33, no. 21, pp. 43-48).

Kelly. A. (2000). *Mobile Robot Localization from Large Scale Appearance Mozaics.* International Journal of Robotics Research (vol. 19, no. 11, pp. 1104-1125).

Enrico Silani, Marco Lovera (2006). *Star Identification Algorithms: Novel Approach & Comparison Study.* IEEE Transactions on Aerospace and Electronic Systems (vol. 42, no. 4, pp. 1275-1288).