Development of New Block Expansion Algorithm for Indoor Mobile Robot Localization

Heesu Roh*, Jeongho Han, Jungwoo Lee,
Kwangwoo Lee, and Soosung Lee
Korea Science Academy
Busan 614-822, Korea
*roh6828@hanmail.net

Daegeun Seo
Pusan National University
Pusan 609-735, Korea

Abstract— **Indoor mobile robots are utilized more and more in modern lives. In order to effectively manipulate their activity, it is important to localize their position precisely. Although a number of different ways using several ultrasonic beacons were developed, these methods are restricted to a relatively small area because all the beacons should be localized accurately before any actual measurement. In this paper, a new methodology was developed to have a robot localized continuously with only three pre-localized beacons. The "block-expansion algorithm" developed in this paper consists of trilateration, auto-calibration and three-point-extraction algorithms, of which the three-point-extraction algorithm is a totally new method. Validity of this method was verified through experiments for two and three triangular areas where the localized path error was confirmed to be less than 3.8% of the robot length in average. In addition, the path error turned out not to accumulate to a considerable extent. The result enables the robot to localize its path by autonomously recognizing the positions of newly appearing beacons, no matter how broad the region is. This algorithm is expected to enable robot localization technology to be more practical for more broad indoor regions.**

Keywords—**localization, trilateration, auto-calibration, three-point-extraction**

I. INTRODUCTION

In the modern life, the application of mobile robots is expanding its scope from cleaning to autonomous security guards, guidance for the elderly and a variety of industrial automation. However, localizing mobile robots has remained a topic of interest for a long time in the process of developing and marketing the mobile robots, since it is the most fundamental and important technique to manipulate the mobile robots [1, 2]. Of a number of prior researches of the robot localization, the method of adapting a number of ultrasonic sensors as "beacons" to localize the up-to-date location of the robot is one of the most widely used techniques [3, 4]. Even though this method has been a topic for numerous related researches due to its high efficiency and low cost, it assumes the investigator to be aware of the locations of all the beacons beforehand [5, 6]. That means that a user should measure the absolute coordinates of all the beacons installed in a room and instruct them to the robot, which is quite prohibiting in a large room. Therefore, the method has a disadvantage of confining its application to a relatively small area [7, 8].

Hence, for application of the mobile robots in a wide region, it is important to develop a method by which we can deductively localize the mobile robot by measuring the absolute location of newly appearing beacons on the basis of existing beacon information [9, 10]. Therefore, in this paper, a new localization algorithm was developed, by which the robot can determine the absolute position of a new beacon based on the information of previously constructed system of beacons and the path tracking history of the robot itself [11-15]. Further localization of the robot is performed by utilizing this new beacon along the other beacons that have been used in the path tracking. The new methodology allows the robot to be localized continuously if it were made to travel from one triangular area to another triangular area. The "block-expansion algorithm" developed in this paper consists of trilateration, auto-calibration and three-point-extraction algorithms, of which the three-point-extraction algorithm is a totally new method. Validity of this method was verified through experiments.

II. LOCALIZATION METHOD

The experimental system was materialized by utilizing four beacons, which geometrically comprised a shape of a parallelogram. Two triangular areas are defined by dividing this parallelogram with respect to the diagonal. The area on which the robot lies at the beginning is called "Area 1", and the other area to which the robot further approaches is called "Area 2". At first, the robot obtains its absolute coordinates by utilizing three beacons within the "Area 1" along with the trilateration algorithm [16-18]. However, when the robot crosses the border between the two triangular regions, "Area 1" and "Area 2", it measures the absolute coordinate of the fourth beacon by the auto-calibration algorithm [19-21]. The algorithms are to be described in Sec. III.

In order to adopt the auto-calibration algorithm, the robot utilizes its three unique positions which are one of the closest positions to the border. As soon as the robot obtains the absolute coordinate of the fourth beacon after reaching "Area 2", it converts one of the three beacons previously used in the trilateration algorithm into the fourth beacon. The new triangular beacon now comes to consist of the fourth beacon and the two beacons at the end of the border line segment between "Area 1" and "Area 2". Even if the number of beacons and areas are increased to five and three, the auto-calibration can be done by the same principle and algorithm.

978-1-4244-1676-9/08 /$25.00

If this algorithm is adopted recursively, the robot can advance as far as possible localizing itself continuously. Every time the robot crosses each border line segment, it can obtain the absolute coordinate of the new beacon by auto-calibrating it at three unique positions near the border. When the robot passes the border, it adopts trilateration by using the new beacon whose absolute coordinate is known by the auto-calibration. Even though the robot, in fact, converts only one of the three beacons used in trilateration for every time it crosses the border, if it repeats the process recursively, it can advance as many areas as possible. Without this recursive method, the use of a mobile robot requires a user to measure the absolute coordinates of all the installed beacons and input them to the robot, which is quite prohibiting in a very large room. However, with the block expansion algorithm, the aforementioned tedious process can be avoided since the only necessary measurement is the absolute coordinate of initial three beacons. Regardless of how large the room is, the robot can autonomously recognize the absolute coordinate of newly appearing beacons thereby independently tracking its own path while moving.

III. BLOCK EXPANSION ALGORITHM

Coordinate calculations in this paper mainly consist of two techniques: trilateration and auto-calibration. Along with them, another mathematical algorithm is needed to determine three unique positions of the robot in the proximity of the borderline between the two triangular areas. The three positions are for auto-calibrating the fourth beacon, for which we have newly developed "three points extraction algorithm".

A. *Trilateration Algorithm*

For further discussion, we assume the situation in which the three beacons comprise one triangular area and each beacon measures the distance between itself and the robot, respectively. From the three distance measurements, the position of the mobile robot is determined as precisely as possible.

If the measurements were conducted ideally precise, the robot would lie where the three circles intersect each other. The three circles are obtained by cutting three spheres whose centers are positions of beacons and radii are d_a, d_b and d_c (distance figures) on the plane that is parallel to the ground and intersects an ultrasonic sensor of the robot. However, the three circles do not intersect each other at one unique position since an actual measurement cannot be ideal. Therefore, the optimal point needs to be determined, which maximizes the existence potential from the probabilistic point of view. The three circles aforementioned produce three unique lines of intersection, which again produces three unique points of intersections with each other. Since the three points do not coincide with each other due to the measurement error, they actually comprise a triangle. The center of gravity of this triangle is used as a probabilistically optimal point.

B. *Auto-Calibration Algorithm*

In this case, the fundamental principle is identical to that of trilateration algorithm except the fact that z-coordinate is no longer fixed (in the previous section, the z-coordinate was the height of an ultrasonic sensor attached to the robot). Our goal is to determine the three-dimensional coordinate of the fourth beacon, which maximizes the potential to coincide with an actual position of the fourth beacon. Also in this case, we can think of three spheres whose centers are the positions of beacons and whose radii are the distance figures. The three spheres intersect each other at two unique positions and only one of them lies above the surface of ground. The only one point of intersection lying above the ground is what is used as the position of the fourth beacon.

C. *Three Points Extraction Algorithm for Auto-Calibration*

The robot needs to determine the position of the fourth beacon by auto-calibration. For the purpose, it is quite important to determine the three points at which the robot measures the distance from the fourth beacon. For this, the user must declare three variables, d_1, d_2 and d_3 in sequence. d_1 is the distance between the boundary and the first robot position for auto-calibration. Similarly, d_2 is the distance between the boundary and the second robot position for auto-calibration. d_3 is used in a similar way within the algorithm. At "Area 1", the robot already knows the coordinates of the three beacons, A, B, and C. Then we can also find the coordinate of the points where the position of beacons A and B are projected to the plane which is parallel to the ground and intersects the robot. Let these two points A' and B'. Then the distance from the robot position to this line segment can found to be A'B', which is denoted as $D(x, y)$.

While the robot is in motion, it saves its current position for the first coordinate for auto-calibration when $D(x, y)$ becomes bigger than d_1 for the first time or becomes smaller than d_1 for the first time. The purpose is to use the point at which the distance from the boundary becomes d_1 for the first time, but since the data is discrete there may not be a point at which the distance exactly becomes d_1. After the robot finds a coordinate for auto-calibration according to the variable d_1, it needs to find another coordinate for auto-calibration according to the second variable d_2. Then the robot applies the same methodology for the third variable d_3 and saves the coordinate for auto-calibration. If the robot conducts auto-calibration with these three points, it can obtain the position of the fourth beacon. This algorithm is depicted as a flow chart given in Fig. 1. In the figure, Δx is the change in x-coordinate between an arbitrary position of a localized path and the position after one time step.

The localization occurs once every 40ms in our system, implying that aforementioned one time step is equal to 40ms. In a similar way, Δy is the change in y-coordinate between the two nearest position data within the localized path.

For better understanding, the flow chart is accompanied by two additional figures, Figs. 2 and 3. Fig. 2 depicts how a position for auto-calibration is determined according to a user-dependent variable d_1 while the robot is in motion. Fig. 3 illustrates an overall experimental system and terminologies of this paper.

Application of the algorithm can be expanded to three areas. For this case also, overall principle and methodologies are the same, except that an additional beacon E needs to be supplemented to comprise three triangular areas.

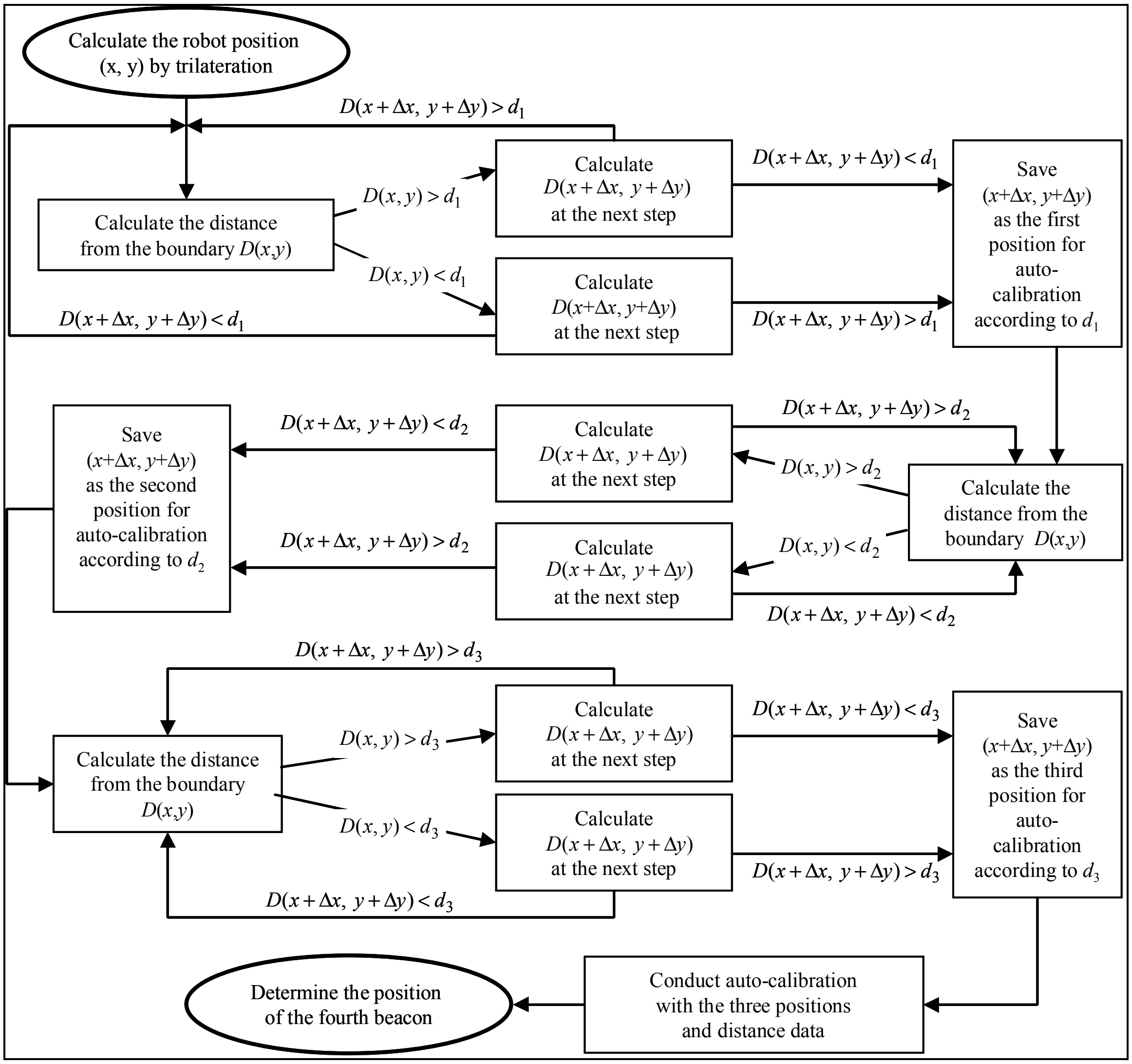

Figure 1. Flow charge of the three points extraction algorithm for auto-calibration.

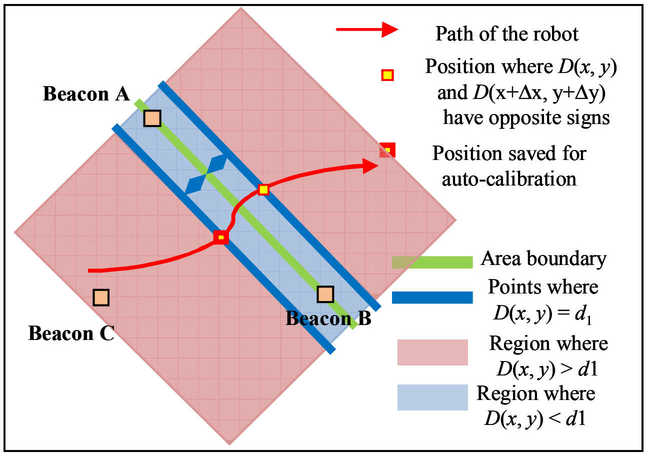

Figure 2. Determination of the points for auto-calibration according to the first variable d_1.

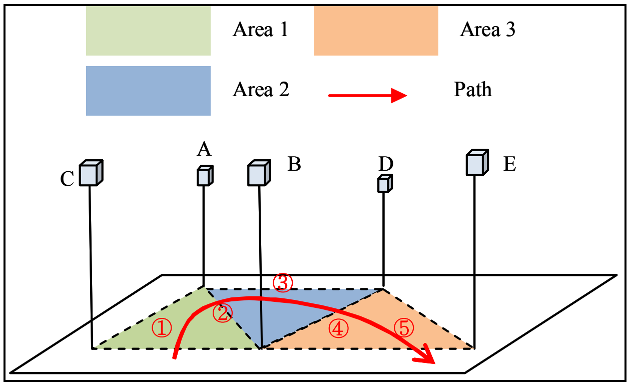

Figure 3. Block expansion algorithm for three areas: ① Trilateration algorithm (A, B, C beacons used), ② Three points extraction algorithm + auto-calibration algorithm (D beacon used), ③ Trilateration algorithm (A, B, D beacons used), ④ Three points extraction algorithm + auto-calibration algorithm (E beacon used), ⑤ Trilateration algorithm (B, D, E beacon used)

IV. EXPERIMENTS AND DISCUSSION

A. *Actual Path of the Robot in Motion*

In order to compare the accuracy of the block expansion algorithm developed in this research, the actual path of the robot in motion needs to be known. In order to obtain the path data, a CCD camera was attached above the floor and the motion of the robot was recorded continuously. During the experiment, the robot was set to move in a parabolic path by driving a system interface program [11-13]. The recorded path of the mobile robot was converted to actual coordinate scale with a computer. Table I shows the initial coordinates of the beacons used in the measurements.

TABLE I. INITIAL COORDINATES OF BEACONS (UNIT: mm)

Beacon	***X***	***Y***	***Z***
A	3467	101	825
B	1,551	139	1,441
C	2,459	1,870	1,457
D	921	2,065	1,341
E	351	132	835

B. *Application of the Algorithm to Two areas*

To extract three points in different ways implies that the magnitudes of d_1, d_2, and d_3 are varied for the extraction. The d_1 was fixed to 450mm, but d_2 and d_2 were subsequently changed to (320mm, 100mm), (250mm, 150mm), (250mm, 50mm), (250mm, 20mm), (210mm, 150mm), (200mm, 110mm). Fig. 4 compares the actual and localized paths for one of the cases. In all the cases, the paths are indeed a curve in the *x*-*y* plane. Thus, we can define a variable σ, which is an RMS difference between the *y*-coordinate of a localized path and that of an actual path. Table II shows how this RMS value σ changes for different values of d_2 and d_3. The two paths differ in the *y*-coordinate by about 33.3mm, which slightly varies according to different values of d_1, d_2 and d_3. The precision can be analyzed in another way. Let σ' be an average distance from every single data point in a localized path to an actual path. Table III shows the variation of the σ' in relation to d_1, d_2, and d_3, where the path difference is 21.9mm in average. Considering that the robot used in the experiment is 450mm long, this much difference corresponds to 4.9% of the robot length in the path estimation. The result confirms the efficacy and validity of the localization algorithm.

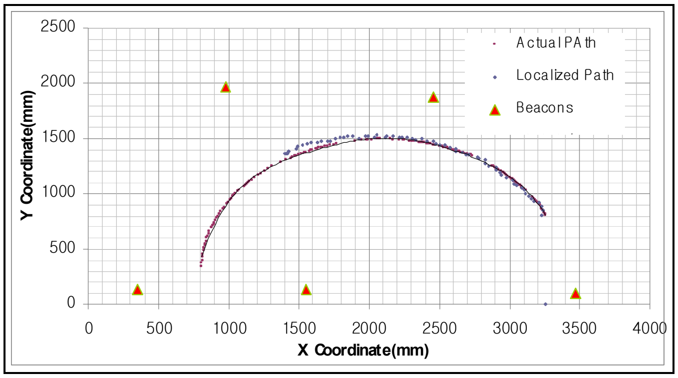

Figure 4. Comparison of an actual path with a localized path in which d_1 = 450mm, d_2 = 320mm, and d_3 = 100mm.

TABLE II. σ FOR DIFFERENT SETS OF d_1, d_2 AND d_3 (UNIT: mm)

d_1	d_2	d_3	σ
450	320	100	44.4
450	250	150	43.4
450	250	50	29.3
450	250	20	31.3
450	210	150	35.4
450	200	110	28.2
Average			33.3

TABLE III. σ' FOR DIFFERENT SETS OF d_1, d_2 AND d_3 (UNIT: mm)

d_1	d_2	d_3	σ'
450	320	100	30.2
450	250	150	29.1
450	250	50	16.3
450	250	20	19.3
450	210	150	22.8
450	200	110	13.9
Average			21.9

C. *Change of the Path Error with respect to the Propagation Distanc: Two Areas*

The path error was investigated how to develop with respect to the distance moved. Experiments were conducted for the six cases in Table III. Fig. 5 is the result of one of the cases.

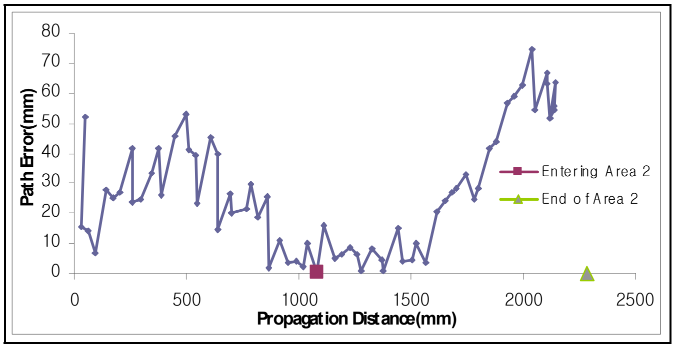

Figure 5. Path error development during the motion. In this case, the path error means the distance from a particular point in a localized path to an actual path. d_1 = 450mm, d_2 = 320mm, and d_3 = 100mm.

After start, the path error abruptly increases, but then decreases if the robot moves more than 500mm. The abrupt increase is considered due to a disturbance in the localization when the robot was accelerating. While the robot moves inside the Area 1 and Area 2, the error remains quite small. However, when the robot approaches the other end of Area 2, the path error again increases rapidly due to the fact that the beacons B, C, and D are getting farther apart from the robot. However, if there were another beacon E outside Area 2 along with Area 3, the error would have not increased to this extent. This gives another reason why the block expansion algorithm is important to reduce the path error.

D. *Application of the Algorithm to Three Areas*

The block expansion algorithm was applied to a wider region, i.e. three areas. Fig. 6 is the result that compares the actual and localized paths of the robot over the three areas. d_4, d_5 and d_6 in the figure are the three new positions for the auto-calibration. Table IV shows the RMS path difference σ for different sets of d_1, d_2, d_3, d_4, d_5 and d_6.

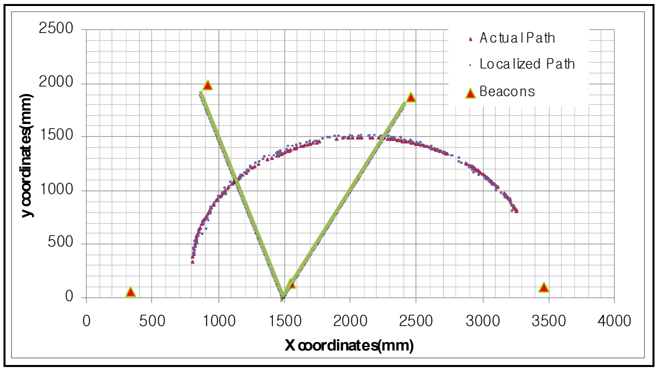

Figure 6. Comparison between an actual path and a localized path; d_1 = 450mm, d_2 = 250mm, d_3 = 50mm, d_4 = 200mm, d_5 = 200mm, and d_6 = 200mm.

TABLE IV. σ FOR DIFFERENT SETS OF d_1, d_2, d_3, d_4, d_5 AND d_6 (UNIT: mm)

d_1	d_2	d_3	d_4	d_5	d_6	σ
450	250	50	200	200	200	34.9
450	250	20	300	400	700	66.9
450	250	20	300	600	750	37.6
Average						46.5

The average value of σ is 46.5mm, which is twice the RMS value with two areas. However, when the distance from a particular point in a localized path to an actual path is calculated, the average value, σ', is as small as 17.3mm as evidenced in Table V. When the block expansion algorithm is applied to three area cases, a particular point in a localized path has an average distance of 17.3mm apart from an actual path.

TABLE V. σ' FOR DIFFERENT SETS OF d_1, d_2, d_3, d_4, d_5 AND d_6 (UNIT: mm)

d_1	d_2	d_3	d_4	d_5	d_6	σ '
450	250	50	200	200	200	14.8
450	250	20	300	400	700	17.9
450	250	20	300	600	750	19.1
Average						17.3

E. Change of the Path Error with respect to the Propagation Distanc: Three Areas

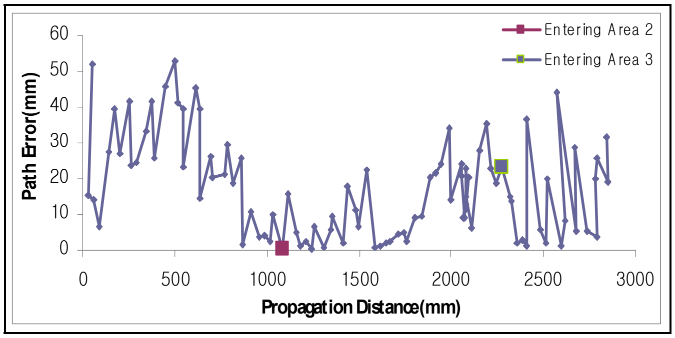

Figure 7. Development of the path error with respect to the travel distance when the block-expansion algorithm is applied to three areas. d_1 = 450mm, d_2 = 250mm, d_3 = 50mm, d_4 = 200mm, d_5 = 200mm, and d_6 = 200mm.

In order to investigate how the path error develops with respect to the propagation distance for three areas, similar experiments to those in Sec. IV-C were conducted for the three cases in Table V.

According to the result in Fig. 7, even though the path error randomly increases or decreases while the robot stays in motion, it is clear that the path error remains less than 50mm, which is of practical significance.

Another method to analyze the path error development by the block-expansion algorithm is to compare the path error data using two areas with that using three areas provided that the two cases use the same d_1, d_2 and d_3. The robot is made to auto-calibrate the beacon D in the same way as that for the two cases. However, in one case, the robot moves without the beacon E and Area 3 while in the other case, the robot executes the block-expansion algorithm one more time.

Fig. 8 shows how the path error develops differently for the localization with two areas and with three areas. The path error develops in almost the same pattern for the two cases. However, when the robot enters "Area 2", the path error increases more rapidly in the two area case. In the three area case, the robot adopts another auto-calibration near beacon E. What is important is that, in the case using three areas, even though the path error fluctuates, its range is still smaller than 50mm at maximum. However, in the case using two areas, the path error abruptly increases at the end to the extent of 70mm. This implies that, by recursively applying the block-expansion algorithm, we have a potential to maintain the fluctuation of the path error smaller than 50mm, which gives great practical significance.

The precision in the localization can be further improved by establishing a more stable way to determine the three points for auto-calibration. Since the path error sensitively changes according to the way the (d_1, d_2, d_3) are set, it is important to investigate a specific method to efficiently establish the three points. For the case with three areas, we also need to investigate (d_4, d_5, d_6).

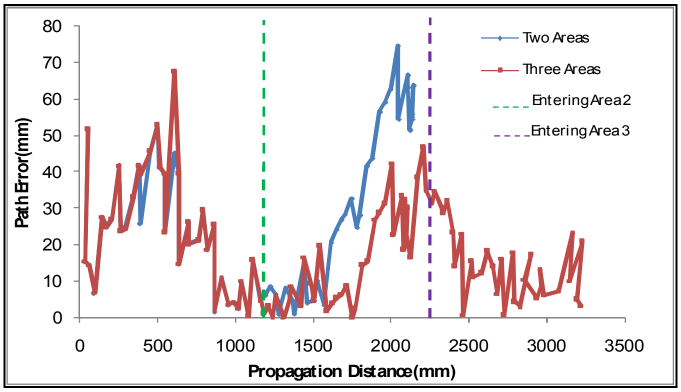

Figure 8. Comparison of the path error of the cases with two areas with that with three areas. d_1 = 450mm, d_2 = 250mm, and d_3 = 20mm.

V. CONCLUSIONS

In order to use beacons in traditional mobile robot localization technology, it is necessary to input all absolute coordinates of beacons to the system. Otherwise, the system can use auto-calibration, but the user needs to input the location of the robot during the auto-calibration. This has been an obstacle in localizing a mobile robot in a broad region.

In this paper, we suggested "block expansion algorithm" in order to resolve this problem. In the algorithm, absolute coordinates of only the initial three beacons are all that the user has to measure. When the robot finds a new beacon, it will autonomously auto-calibrate the beacon at three distinct positions near the boundary; during this process, the robot does not stop localizing itself. Therefore, even if the robot escapes the former region and enters a new region consisting of the new beacon, it still can localize its path since it has obtained the coordinate of the new beacon. An entire "block-expansion algorithm" is mainly composed of "trilateration algorithm", "auto-calibration algorithm", and "three points extraction algorithm" for the auto-calibration. Of these, the three points extraction algorithm has been developed newly in this paper.

The validity of the developed algorithm was verified through experiments. Two triangular areas were defined by arranging four beacons in parallelogram. When the robot entered a new triangular area, it auto-calibrated the fourth beacon autonomously. Then it converted one of the three beacons used in the localization into the fourth beacon, which enabled continuous localization even when the robot escaped the initial triangular area. In order to evaluate the precision of the localization, the RMS value of y-coordinate differences was estimated between a localized path and an actual path. The distance between a particular point and an actual path was measured to be 21.9mm in average.

The same method was applied to a further wider region, three areas, to check its flexibility and expandability. The three area region comprised five beacons. The RMS value of y-coordinate differences turned out to be 46.5mm, and the average value of distances from points in a localized path to an actual path turned out to be 17.3mm, respectively. Considering that the robot used in the experiments is 450mm long, the difference of 17.3mm corresponds to 3.8% of the robot length, which is quite small and thus confirms the accuracy and efficacy of the localization algorithm developed in this paper. Further improvement of the localization precision is considered to be possible by establishing a more stable way to determine the three points for auto-calibration.

The result of this paper enables the robot to localize its path by autonomously recognizing the positions of newly appearing beacons, no matter how broad the region is. The only thing the user has to do at the initial stage is to measure the positions of first three beacons. This algorithm is expected to enable robot localization technology to be more practical for more broad indoor regions.

REFERENCES

[1] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte-Carlo localization for mobile robots," Artificial Intelligence, vol. 128, pp. 99-141, 2001.

[2] W. Hu, T. Downs, G, Wyeth, M. Milford, and D. Prasser, "A modified particle filter for simultaneous robot localization and landmark tracking in an indoor environment," Proceedings of IEEE International Conference on Robotics and Automation, pp. 1-7, 2004.

[3] W. Burgard, D, Fox, and S. Thurn, "Active mobile robot localization," Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, pp. 201-208, 1997.

[4] R. Negenborn, "Robot Localization and Kalman Filter," MS thesis, Utrecht University, 2003.

[5] C. Detweiler, "Passive mobile robot localization within a fixed beacon field," MS thesis, Computer Science and Artificial Intelligence Laboratory, MIT, 2006.

[6] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," Journal of Artificial Intelligence, vol. 11, pp. 391-427, 1999.

[7] A. Singhal, Issues in Autonomous Mobile Robot Navigation, Monograph of University of Rochester Press, Rochester, 1997.

[8] I. J. Cox and G. Wilfong, Autonomous Robot Vehicles, Springer-Verlag, New York, 1990.

[9] S. Roumeliotis, "Reliable Mobile Robot Localization," Ph.D. thesis, University of Southern California, 1999.

[10] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A probabilistic approach to collaborative multi-robot localization," Autonomous robotics on Heterogeneous Multi-Robot Systems, vol. 8, pp. 325-344, 2000.

[11] T. Jin, and J. Lee, "Localization and navigation of a mobile robot using single ultrasonic sensor module," Journal of the Institute of Electronics Engineers of Korea, vol. 42, 2005.

[12] T. Jin, and J. Lee, "Position estimation of autonomous mobile robot using geometric information of a moving object," Journal of Korean Fuzzy Logic & Intelligent System Society, vol. 24, pp 438-444, 2004.

[13] S. Kim, D. Lee, and J. Lee, "Indoor Localization Scheme of a Mobile Robot Applying RFID Technology," Journal of Control, Automation, and Systems Engineering, vol. 11, 2005.

[14] S. Han, H. Lim, and J. Lee, "An Efficient Localization Scheme for a Differential-Driving Mobile Robot Based on RFID System," IEEE Trans. Ind. Electron., vol. 54, pp. 3362-3369 , 2007.

[15] D. Seo, S. Cho, and J. Lee, "Localization algorithm for a mobile robot using iGS," J. Korean Ins. Control, Robotics and Systems, vol. 14, pp. 242-247, 2008.

[16] J. Borenstein, H. R. Everett, and D. Wehe, "Mobile robot positioning: sensors and techniques," J. Robot Systems, vol. 14, pp. 231-249, 1998.

[17] F. Thomas and L. Ros, "Revisiting trilateration for robot localization," IEEE. Trans. Robotics, vol. 21, pp. 93-101, 2005.

[18] L. Kleeman, "Optimal Estimation of Position and Heading for Mobile Robots Using Ultrasonic Beacons and Dead-reckoning," IEEE Int. Conf. Robotics and Automation, 1992, pp. 2582—2587.

[19] E. M. Foxlin, "Generalized architecture for simultaneous localization, auto-calibration, and map-building," Proc. IEEE Int. Conf. Robots and Systems, 2002, pp. 3154-3160.

[20] P. Weckesser, R. Dillmann, M. Elbs, and S. Hampel, "Multiple sensor processing for high-precision navigation and environmental modeling with a mobile robot," **Proc.** IEEE Int. Conf. Robots and Systems, 1995, pp. 453—458.

[21] C. C. Tsai, H. H. Lin, and S. W. Lai, "Multisensor 3D Posture Determination of a Mobile Robot Using Inertial and Ultrasonic Sensors," Journal of Intelligent and Robotic Systems, vol. 42, 317-335, 2005.