Increasing SLAM Performance by Integrating Grid and Topology Map

Suhyeon Kim
Mokwon Univ./Intelligent Robot Eng., Daejeon, Korea
Email: dukesoo@mokwon.ac.kr

Hyungrae Kim and Tae-Kyu Yang
Mokwon Univ./Intelligent Robot Eng., Daejeon, Korea
Email: {khr003, tkyang}@mokwon.ac.kr

Abstract— **The technique of simultaneous localization and mapping is the most important research topic in mobile robotics. In the process of building a map in its available memory, the robot memorizes environmental information on the plane of grid or topology. Several approaches about this technique have been presented so far, but most of them use mapping technique as either grid-based map or topology-based map. In this paper we propose a frame of solving the SLAM problem of linking map covering, map building, localizing, path finding and obstacle avoiding in an automated way. Some algorithms integrating grid and topology map are considered and they make the SLAM performance faster and more stable. The proposed scheme uses an occupancy grid map in representing the environment and then formulates topological information in path finding by Dijikstra algorithm. The mapping process is shown and the shortest path is decided on grid based map. Then topological information such as direction, distance are calculated on a simulator program then transmitted to robot hardware devices. The localization process and the dynamic obstacle avoidance can be accomplished by topological information on the grid map. While mapping and moving, pose of the robot is adjusted for correct localization by implementing additional pixel based image layer and tracking features. A laser range finder and electronic compass systems are implemented on the mobile robot and DC geared motor wheels are individually controlled by PD based adaptive way. Simulations and experimental results show performance and efficiency of the proposed scheme.**

***Index Terms*— SLAM, hybrid map, path finding, Dijikstra algorithm**

I. INTRODUCTION

A. Basics and research background

Mobile robot research has been conducted for some decades because it has simple kinematic or dynamic structure and is also based mainly on easy control techniques. A mobile robot should know its destinations or trajectories of the course to do some given works or to decide a more efficient way. With some sensors on the board, the robot can localize or recognize its position by mapping available routes in its memory database.

Map buildings depend strongly on the characteristics of the sensors that provide the raw data. In order to create a map using sensors, such as ultrasonic range measurements, laser range finder, etc. several methods are used such as sensor interpretations, integration over time, pose estimation, global grid building and exploration. Most available mapping methods are metric grid map or topology map[1].

In an occupancy grid, the environment is tessellated by discrete grids, where each cell is either filled or empty. This method is of particular value when a robot is equipped with range-based sensors because the range values of each sensor, combined with the absolute position of the robot, can be used directly to update the filled or empty value of each cell. There remain two main disadvantages of the occupancy grid approach. First, the size of the map in robot memory grows with the size of the environment and if a small cell size is used, this size can quickly become untenable. This occupancy grid approach is not compatible with the closed world assumption, which enabled continuous representations to have potentially very small memory requirements in large, sparse environments. In contrast, the occupancy grid must have memory set aside for every cell in the matrix. Furthermore, any fixed decomposition method such as this one imposes a geometric grid on the world a priori, regardless of the environmental details. This can be inaccurate in cases where geometry is not the most salient feature of the environment.

For these reasons, an alternative, called topological decomposition, has been the subject of some exploration in mobile robotics. Topological approaches avoid direct measurement of geometric environmental qualities, instead concentrating on characteristics of the environment that are most relevant to the robot for localization. The topological representations have three key advantages over grid-based approaches: they permit fast planning, facilitate interfacing to symbolic planners and problem-solvers and provide natural interfaces for human instructions. Moreover, it is easier to generate and maintain global consistency for topological maps than for metric maps. Since topological approaches usually do not

require exact determination of the geometric position of the robot, they often recover better from drift and slippage that must constantly be monitored and compensated in grid based approaches. However, these techniques often ignore valuable metric information and they still lack the ability to describe individual objects in an environment. In addition, coherent topological maps are often difficult to learn and maintain in large scale environments. One of the hardest problem in mapping is the perceptual aliasing, which is a problem of the data association, also known as the closing the loop or the revisiting problem[2]. With sufficient rich sensory information and features abstraction, detecting unique sensing signatures that describe such place will easily solve this problem. Topology based maps are fairly robust against sensor noise and small environmental changes, and have rich computational properties. Occupancy grids have been implemented with laser range finders, stereo vision sensors and even with a combination of sonar, infrared sensors and sensory data obtained from stereo vision.

The Simultaneous Localization and Map building i.e. SLAM is considered by assuming independence between robot's positions and features locations. SLAM is the work of making the exact environments and maps while rolling away the place without any special sensors from exterior systems. This technique is basic for mobile robot but there have been few successful results yet because exact localization of robot's position is also related to exactness of the map and its memories are limited[3].

B. Outline

In this study we build an autonomous mobile robot which has simultaneous mapping ability, localization, path finding and dynamic obstacle avoiding. The robot system has a simple laser range finder in front and an electronic compass. Cores of our research are hybrid mapping technology which integrate grid based map and topology based map. Building map and coverage process map is made by backtracking algorithm by robot sensing and transmitting its environmental information to our simulator program. After building a map, from the predefined information of starting point, passage points and goal point, the shortest path can be drawn on simulator window by the Dijkstra algorithm. Another automated algorithm picks topological information from graphical path on grids and then gives movement command of direction angle and distance to next passage point to robot hardware devices, i.e. ATmega128 controller, DC geared motors and sensor system.

We developed more exact but simply structured and cost-saving algorithms which can acquire environmental map with a laser range finder. The hybrid mapping process is automated by minimum operation and sensor information are hypothesized then adaptively controlled for more exact localization. After mapping and localizing process, we apply some modified Dijkstra algorithm which can be used to find the fastest way to some objective points. On simulations some algorithms for optimal paths are tried and then corresponding experiments are carried out in the indoor office.

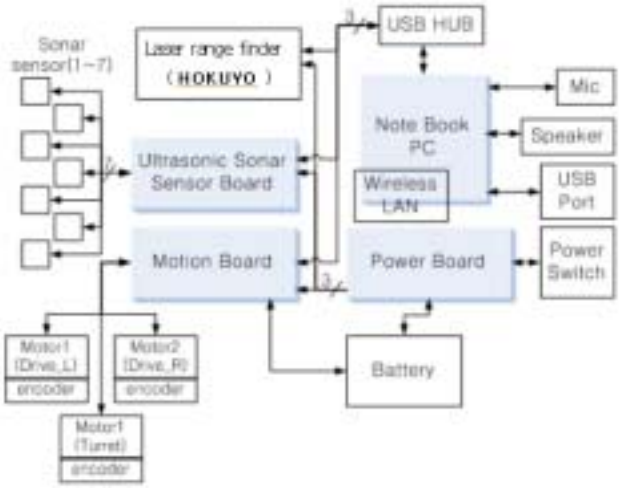

Fig.1 A schematic of the mobile robot

II. RELATED WORKS

Castellanos et al.[4] use the symmetries and perturbations map i.e. SPmap. SPmap suboptimal estimation is performed by application of the EKF while considering measurements and observations gathered by sensors mounted on the mobile robot. This model emphasizes the role of the covariance matrix as a memory of the relations between features included into the state vector at different time instants. Namely, whenever the mobile robot returned to areas of the environment already visited and learned, the system could re-estimate the location of features in the map which were not visible from the current mobile robot location. At the same time, they were related to current observations through the off diagonal elements of the system covariance matrix[5]. But, this performance is limited in the aspect of computation time and memory as the estimated features increase.

A popular probabilistic framework for position tracking is the method of applying *Kalman filters.* Kalman filter-based method represents its belief of the robot's position by a unimodal Gaussian distribution over the three-dimensional state-space of the robot. The mode of this distribution yields the current position of the robot, and the variance represents the robot's uncertainty. Whenever the robot moves, the Gaussian is shifted according to the distance measured by the robot's odometry. Existing applications of Kalman filtering to position estimation for mobile robots are similar in how they model the motion of the robot. They differ mostly in how they update the Gaussian according to new sensory input.

The advantage of Kalman filter-based techniques lies in their efficiency and in the high accuracy that can be obtained. The restriction of a uni-modal Gaussian distribution, however, is prone to fail if the position of a robot has to be estimated from scratch, i.e. without knowledge about the starting position of the robot. Furthermore, these techniques are typically unable to recover from localization failures. Markov localization, which has been employed successfully in several variants overcomes the disadvantage of Kalman filter based

techniques. The different variants of this technique can be roughly distinguished by the type of discretization used for the representation of the state space. Some use Markov localization for landmark-based navigation, and the state space is organized according to the topological structure of the environment. Here, nodes of the topological graph correspond to distinctive places in hallways such as openings or junctions and the connections between these places. The vast majority of existing approaches to localization differ from ours in that they address localization in static environments. Therefore, these methods are prone to fail in highly dynamic environments in which, for example, large crowds of people surround the robot. However, dynamic approaches have great practical importance, and many envisioned application domains of service robots involve people and populated environments[6]-[7].

FastSLAM is another stochastic system using Rao-Blackwellized particle filter which differs from existing approaches in that it exploits sparsity in the dependencies between data and the state variables over time to factor the SLAM problem into a set of low dimensional problems[5]. It samples over the robot's path and data associations and computes independent landmark estimates conditioned on each particle. FastSLAM has several advantages over EKF based approaches to the SLAM problem such as logarithmic complexity, multi hyp0othesis data association and convergence without the full covariance matrix. But FastSLAM also has some limitations. FastSLAM ignores correlation information which will inevitably cause it to underestimate the covariance of landmarks in the map and this may makes the process of adding new landmarks more difficult. Another defect is loop closing problem. Initially FastSLAM's multi hypothesis approach seemed well suited to solving the data association problem induced by the large pose uncertainty at the end of the loop. But throwing away correlation slows down the rate at which observations can affect the positions of other landmarks thus slowing the convergence of the map.

There have been two categories about localization for mobile robot, of which one used grid map and the other uses topological map. The method of using grid map divides whole areas as the regular small grids and the rate of each grid is occupied and represented in probability value. Compared with the other methods, topological map method using grid map shows the whole environment information and this information can be used to find the fastest or optimal path for the robot[8].

Generalized Voronoi Graph (GVG) builds a topology map and forms path of same distance from the environment or obstacles and the robot uses distance information measured from sensors. Because this method can be used directly without additional data processing, mapping can be possible for various complicated environments and expansion of the map is easy. But it can be influenced by sensor noises and unnecessary nodes. To overcome such difficulties, GVG method was supplemented by thinning algorithm [9].

III. MATHEMATICAL CONSIDERATIONS

A. Kinematic and Dynamics analysis

Our mobile robot has two differential wheels driven by DC geared motors and a ball caster supporting the weights while allowing directional change. Velocities of the robot in each axis direction can be represented as matrix equations.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\phi & -d\sin\phi \\ \sin\phi & d\cos\phi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{1}$$

In equation (1), v is the central velocity of gravity. It shows the relation of control input to velocity. Angular velocity of each wheel and linear velocity vector can be defined as next equation from its kinematical mechanism.

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{L} & -\frac{r}{L} \end{bmatrix} \begin{bmatrix} w_R \\ w_L \end{bmatrix} \tag{2}$$

From the above two equations we can derive next two equations where one is the rectangular coordinate values resulting from angular velocity of each wheel and the other is the value when the center of gravity is located on the extension line of its area center.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\cdot\cos\phi - \frac{r}{L}\cdot d\cos\phi & \frac{r}{2}\cdot\cos\phi + \frac{r}{L}\cdot d\sin\phi \\ \frac{r}{2}\cdot\sin\phi + \frac{r}{L}\cdot d\cos\phi & \frac{r}{2}\cdot\sin\phi - \frac{r}{L}\cdot d\cos\phi \\ \frac{r}{L} & -\frac{r}{L} \end{bmatrix} \begin{bmatrix} w_R \\ w_L \end{bmatrix} \tag{3}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\cdot\cos\phi & \frac{r}{2}\cdot\cos\phi \\ \frac{r}{2}\cdot\sin\phi & \frac{r}{2}\cdot\sin\phi \\ \frac{r}{L} & -\frac{r}{L} \end{bmatrix} \begin{bmatrix} w_R \\ w_L \end{bmatrix} \tag{4}$$

Using above equations in the robot model, we can try various simulation preliminary to experiments.

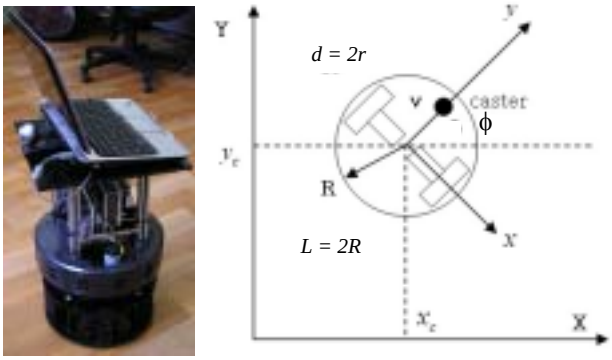

Fig. 2 Differential drive wheeled mobile robot

B. Dynamics and Control

The position of the mobile robot in the global frame X-Y can be defined by the position of the mass center of the mobile robot system, denoted by x-y. X-y is the center of the mobile robot gear, and the angle between the robot's local frame x_c-y_c and the global frame. The kinetic energy of the whole structure is given by the following equation.

$$T = T_l + T_r + T_{kr} \tag{5}$$

T is the kinetic energy that is the consequence of pure translation of the entire vehicle, T_r is the kinetic energy of the rotation of the vehicle in the XY plane, and T_{kr} is the kinetic energy of rotation of the of the motors' wheels. The values of energy terms introduced can be expressed by Equations (6)-(8).

$$T_l = \frac{1}{2} M v_c^2 = \frac{1}{2} M (\dot{x}_c^2 + \dot{y}_c^2) \tag{6}$$

$$T_r = \frac{1}{2} I_A \dot{\theta}^2 \tag{7}$$

$$T_{kr} = \frac{1}{2} I_0 \dot{\theta}_R{}^2 + \frac{1}{2} I_0 \dot{\theta}_L{}^2 \tag{8}$$

M is the mass of the entire vehicle, v is the linear velocity of the vehicle's center of mass, I_A is the moment of inertia of the entire vehicle considering point A, point A is the angle that represents the orientation of the vehicle (Fig. 2), I_0 is the moment of inertia of the rotor/wheel complex and $\dot{\theta}_R$, $\dot{\theta}_L$ are the angular velocities of the right and left hand wheels respectively. Further, the components of the velocity of the point A can be expressed in terms of $\dot{\theta}_R$, $\dot{\theta}_L$.

$$\dot{x}_A = \frac{r}{2}\left(\dot{\theta}_R + \dot{\theta}_L\right)\cos\theta \tag{9}$$

$$\dot{y}_A = \frac{r}{2}\left(\dot{\theta}_R + \dot{\theta}_L\right)\sin\theta \tag{10}$$

$$\dot{\theta} = \frac{r\left(\dot{\theta}_R - \dot{\theta}_L\right)}{2R} \tag{11}$$

D is the distance between points A and C and from above equations and substitute into Lagrange Equations as follows,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_R}\right) - \frac{\partial L}{\partial \theta_R} = \tau_R - K\dot{\theta}_R \tag{12}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_L}\right) - \frac{\partial L}{\partial \theta_L} = \tau_L - K\dot{\theta}_L \tag{13}$$

Here, subscript R and L are right and left actuation torques and Kd_R/dt and Kd_L/dt are the viscous friction torques of right and left wheel-motor systems, respectively. Finally, the dynamic equations of motion can be expressed as, [4]

$$A\ddot{\theta}_R + B\ddot{\theta}_L = \tau_R - K\dot{\theta}_R \tag{14}$$

$$B\ddot{\theta}_R + B\ddot{\theta}_L = \tau_L - K\dot{\theta}_L \tag{15}$$

Where

$$A = \left(\frac{Mr^2}{4} + \frac{\left(I_A + Md^2\right)r^2}{4R^2} + I_0\right) \tag{16}$$

$$B = \left(\frac{Mr^2}{4} - \frac{\left(I_A + Md^2\right)r^2}{4R^2}\right) \tag{17}$$

C. Control system for positioning exactness

Due to nonlinearities and uncertainties such as slips on the ground surface, friction between the wheels and the floor, backlash in motor gears and some interior factors, control performance became worse and the position error tend to be increased according to the locomotion distance. To overcome such setbacks we have designed a PD based adaptive control algorithms. This algorithm compensates errors by analyzing the current position and control angles of each motor from some features and the memorized map. It is partially similar to the fuzzy logic controller which uses a fuzzy set and linguistic rules and it also includes the characteristics of *Kalamn filter* such as the application of previous error information to the current control gain. Begun et al.[10] implemented some fuzzy logic control in localization of their mobile robot. We had developed the modified fuzzy logic control in order to improve positioning accuracies and control performance by utilizing Begun at a1.'s achievement.

Wheel encoders give an odometric measure of the vehicle location. When the original position O in the first driving has changed to the point P, we say that the robot's position is far from exact position by over-rightly in amount of the radial distance. Fig. 3 shows this error position on the plane.

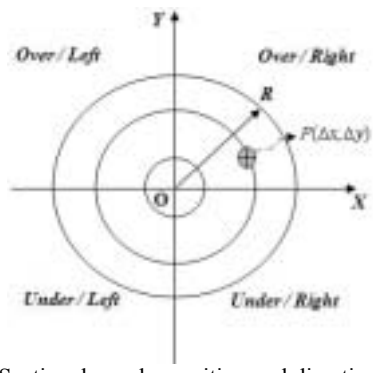

Fig. 3 Sectional area by position and direction error

For the next motion plan and compensation of position error, two wheels' control angles are represented respectively.

$$\theta_L = \theta_L^F + \Delta\theta_L$$
$$\theta_R = \theta_R^F + \Delta\theta_R \quad (19)$$

In the above, θ_L^F and θ_R^F are control angles from current to the posterior position and $\Delta\theta_L$ and $\Delta\theta_R$ are compensation angle reflecting the previous motion variables and current position error. Compensation angles have the component of some factors such as,

$$\Delta\theta_{L,R} = \Delta\theta_{L,R}^P + \Delta\theta_{L,R}^E \quad (20)$$

$$\Delta\theta_{L,R}^P = k_1\theta_{err} + k_2 l_{err} \quad (21)$$

$\theta_{err} = (\theta_m - \theta^P)$, $l_{err} = sign(\Delta x)\cdot(l - 4b)$ are satisfied respectively where l is the straight distance to the next passage point and b is the length of the robot. We have designed the above equations by considering some experimental data which forward directional error would change on the basis of four times longer distance than the robot's length.

$\Delta\theta_{L,R}^E$ is the compensation angle using current position which is decided by the following way and $\Delta\theta_{L,R}^E$ is also controlled angle from the sectional area and the error distance information.

$$\Delta\theta_{L,R}^E = a\cdot Z + b\cdot\frac{|\Delta x|}{\sqrt{\Delta x^2 + \Delta y^2}} \quad (22)$$

$$Z = \eta(x^2 + y^2) + 1 \quad (23)$$

Here, the variable η is updated in each time step as the following iterated relation.

$$\eta_t = \left(1 + \frac{\sqrt{(\Delta P_t)^2 + (\Delta\theta_t)^2} - \sqrt{(\Delta P_{t-1})^2 + (\Delta\theta_{t-1})^2}}{\sqrt{(\Delta P_{t-1})^2 + (\Delta\theta_{t-1})^2}}\right)\eta_{t-1} \quad (24)$$

In equation (22) the variable a and b represent the directional information of the two wheels, front-rear and left-right information is determined.

Table 1 Correction factors on each areal section

	O/R		O/L		U/L		U/R	
	a	b	a	b	a	b	a	b
L	-1	0	-1	1	1	1	1	0
R	-1	1	-1	0	1	0	1	1

While analyzing the localization error, the control forces are exerted in proportion to the distance from the original center and the directional error information from the forward direction are added to next motion plan. This increases the ability of adaptability by updating the control force distribution for compensating the kinematic error in each time step. Fig. 4 shows the contour plot of the error compensation function (23). The parameter η deforms the contour and regulates the proportionality on the error weighting.

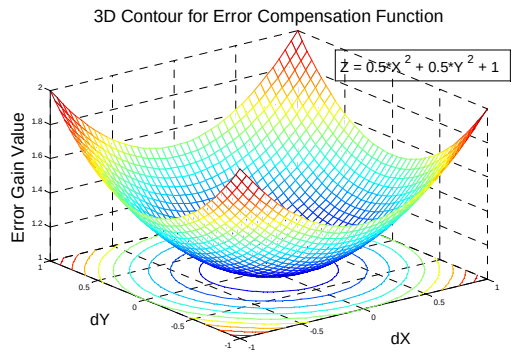

Fig. 4 Contour plot of the error compensation function ($\eta = 0.5$)

IV. SLAM

A. Map building and Coverage pattern

A standard, lossless form of appropriately opportunistic decomposition is termed exact cell decomposition. This method achieves decomposition by selecting boundaries between discrete cells based on geometric criticality. The map representation tessellates the space into areas of free space. The representation can be extremely compact because each area is actually stored as a single node.

Exact decomposition is a function of the particular environment obstacles and free space. If this information is expensive to collect or even unknown, then such approach is not feasible. An alternative is fixed, in which the world is tessellated, transforming the continuous real environment into a discrete approximation for the map. The key disadvantage of this approach stems from its inexact nature. It is possible for narrow passageways to be lost during such transformation.

In this study a map is built by grid based method then processes of path finding, localization and correction are performed by topology based method. The robot has scan type ultrasonic sensor and two same sensors fixed on its sides. To cover all area in some environment, special coverage pattern is needed. This is the problem of coverage pattern and there has been much coverage pattern algorithms research, however each method has its own defects and merits in each environment[11]. Here the backtracking algorithm is applied to cover all grid cells and this is represented on the simulator window real time.

Backtracking algorithm is a form of recursive process which systematically searches for a solution to a problem among all available options. It does so by assuming that the solutions are represented by vectors (v_1, ..., v_m) of values and by traversing, in a depth first manner, the domains of the vectors until the solutions are found. When invoked, the algorithm starts with an empty vector. At each stage it extends the partial vector with a new value. Upon reaching a partial vector (v_1, ..., v_i) which can't represent a partial solution, the algorithm backtracks by removing the trailing value from the vector, and then proceeds by trying to extend the vector with alternative values. Pseudo codes are expressed as follows.

Pseudo Code of the Backtracking Algorithm

```
try(v1,...,vi)
  IF (v1,...,vi) is a solution THEN RETURN (v1,...,vi)
  FOR each v DO
    IF (v1,...,vi,v) is acceptable vector  THEN
      sol = try(v1,...,vi,v)
      IF sol != () THEN RETURN sol
    END
  END
  RETURN ()
```

If S_i is the domain of v_i, then $S_1 \times ... \times S_m$ is the solution space of the problem. The validity criteria used in checking for acceptable vectors determines what portion of that space needs to be searched, and so it also determines the resources required by the algorithm. The traversal of the solution space can be represented by a depth-first traversal of a tree. The tree itself is rarely entirely stored by the algorithm in discourse, instead just a path toward a root is stored to enable the backtracking as you can see from Fig. 3[8].

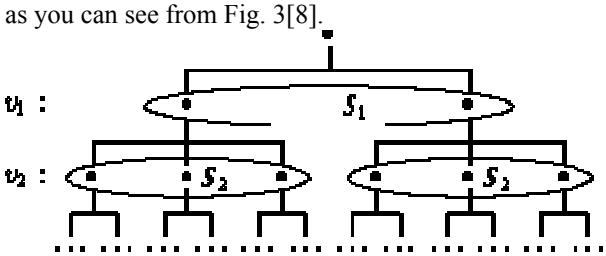

Fig. 5 Tree node representation of backtracking

When using the backtracking algorithm, some occasions came across such that robot should pass the route once it had passed by to escape from isolated regions or blind alleys. At this time path from isolated point to exit point is decided by modified A* algorithm i.e. Dijikstra algorithm[12]-[13]. So some special programming loops are added and at that time any sensor information are not needed for locomotion.

B. Localization

Because of odometry in driving mechanism, sliding effect on the plane and hardware control error, the position or direction has difference to the targeted location. To correct these differences a robot should hypothesize its built map and current location information from sensors. Our robot solves these problems by using simple scan type laser range finder, two ultrasonic sensors in the side and electronic compass on the top.

Firstly, distances from current position are estimated from sensor values analysis and then compared to geometry around the already built grid map. In this time the FastSLAM2.0 algorithm[5] are used for expanding the global map from the estimated local maps. From three distance values and direction information we can transform this information to x and y error values in rectangular coordinate by triangulation method[14]-[17]. The FastSLAM2.0 proposal distribution is designed to generate new robot pose that also agrees with the current observation z_t.

$$s_t^m \sim p\left(s_t \mid s_{t-1}^m, u_t, z_t, n_t\right) \tag{25}$$

Here s_{t-1}^m is the path up to time t-1 attached to the m*th* particle and u_t , z_t , n_t means the most recent control input, sensor measurement and its data association at time t individually.

If the robot's motion model is nonlinear, observations cannot be incorporated into the proposal in closed form. FastSLAM2.0 solves this problem by linearization of the motion model. Drawing a sample from the new proposal distribution given each particle in S_{t-1} is a next three step process.

1. Project the previous pose of the robot s_{t-1}^m forward according to the linear Gaussian motion model. The uncertainty of the resulting pose $\hat{s}_t$ will be a Gaussian.
2. Consider this Gaussian to be the prior distribution of an EKF estimating the pose of the robot. Incorporate the observation z_t into this EKF using the standard EKF updated equations.
3. Draw a sample s_t^m from the resulting posterior distribution.

The main characteristics of human made indoor environments, namely parallelism and perpendicularity between the walls of the room, were preserved in the estimated features. Laser edge is obtained by considering all the available laser points and its covariance matrix is calculated just from the endpoints to avoid optimistic estimations of uncertainty. Corners are defined by two consecutive segments in the robot's environment. Typical indoor environments are formed by perpendicular walls, thus most of the corners to be found by sensors of the mobile robot, will be 90 degree corners. Nevertheless, our discussion is completely general, valid for any corner detected in the environment of the vehicle.

Homogeneous regions are considered as suitable representations of the local environment of the mobile robot, while spurious regions are discarded from further processing. At this stage, homogeneous regions are subdivided into groups of laser points which probably can be approximated by a unique straight line segment. An iterative line fitting algorithm is adapted to plane range finder data processing. The algorithm might be summarized as follows[6].

1. Let P_a and P_b be the points of the starting and ending of a homogeneous region and let x_{Pa} and x_{Pb} be their location vectors with respect to the reference frame of the laser sensor. Calculate the analytical solution for the reference frame attached to the edge E defined by P_a and P_b.
2. Compute an initial estimation of the covariance matrix of the edge C_e by integrating the extreme points P_a and P_b using suboptimal estimation based on the EIF with measurement equation.
3. For all the intermediate points P_i within P_a and P_b whose location vector with respect to the sensor L is given by x_{Pa} , calculate the squared *Mahalanobis* distance D_i^2 from the point P_i to the edge E.
4. If $\max(D_i^2) \le \chi_{r,\alpha}^2$ $\forall i \in$ [S,E], where $\chi_{r,\alpha}^2$ is a threshold value, obtained from the χ^2 distribution, such that the probability of false positive is α and r = 1, then the laser point P_i may be considered coincident with the

edge E up to the symmetries of the pairs, and the process is stopped, otherwise the laser point P_i is labeled as a breakpoint P_b.

5. Detection of a breakpoint P_b triggers the computation of residuals. Whenever partitioning the region (P_s, P_a) at the breakpoint P_b introduces an improvement in the interpretation of the local environment of the mobile robot, that is,

$$\sum_{k=Pa}^{Pe} D_{ae,k}^2 > \sum_{k=Pa}^{Pb} D_{ab,k}^2 + \sum_{k=Pb}^{Pe} D_{be,k}^2 \tag{26}$$

The algorithm recurs by considering the two new regions (P_s, P_b) and (P_b, P_e). Residual verification avoids considering spurious laser readings, undetected by the tracking like algorithm, as breakpoint in the scanned sequence. Fig. 6 shows the process of correcting its error position in counter way by recursive matching of translation and rotation.

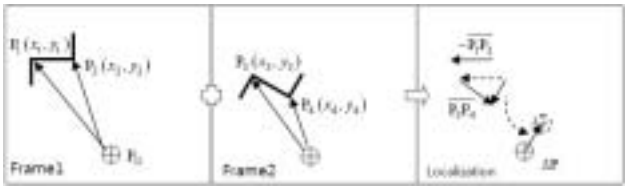

Fig. 6 Localization by featured information

C. Path finding and Obstacle avoidance

To fulfill the special objective or reach the goal in fastest way a path plan is needed. In this study we use the Dijikstra algorithm which calculates the cost function and selects the minimum value. To compare performances, other algorithms are tried in our simulations such as convex hull, Sklansky, STRIPS, etc.[18] After choosing path points, the fastest and most safe way can be found by linking each point. The robot can travel on the line of this curve and each device is controlled by PD based adaptive method. The shortest path needs a starting point, passage points and goal point. Results are graphically represented on built grid map on simulator window. From continuous cell graph we can acquire topological information which is useful to robot's motion. Based on starting point and next passage point, first commands to robot are its initial position, direction and distance to move. In simulator program each point can be searched from next ways.

$$\frac{y_{i+1} - y_i + \varepsilon}{x_{i+1} - x_i + \varepsilon} \neq \frac{y_i - y_{i-1} + \varepsilon}{x_i - x_{i-1} + \varepsilon} \tag{19}$$

In this equation, i means each step and ε is used to prevent infinity overflow error when a denominator approaches zero. From absolute direction information of robot by electronic compass, it is also possible to turn to exact direction at any point. Topological commands give distance and direction to robot. In moving to the next passage point, robot can meet with a dynamic obstacle such as a human, another robot or a closed door. As mentioned above, the problem of where the robot changes its direction depends on where less occupied obstacles are existed. The robot moves closely to side obstacle by using its side sensor which is the fastest way to turn over some obstacle in a plane. While turning, the current position of the robot is graphically updated in a grid cell and compared to original path in each step. If the original path cell can be met with robot position, this means that robot can successfully avoid that obstacle and it is allowed to go ahead on the original path also after turning to next passage point by electronic compass.

V. SIMULATION AND EXPERIMETAL RESULTS

At the start of searching environment, our robot travels from base to far-away end of its memory map. The memory map is the 3 dimensional arrays, preceding 2 arrays element grid element numbers are defined and in the last element relative distance information is included. Acquiring the full information of fixed region, the robot can expand its memory map while traveling to some location. But then previous region is erased in the amount to expanded region.

Our simulator program is made by MATLAB 7.5 and it communicates by serial line to ATmega128 controller which is connected to sensors and actuators. The experiment was accomplished in a Laboratory which composed of desks, partitions and bookcases. Fig. 7 shows the process of map building from an unknown area and the finished result of map building. Fig. 8 shows the shortest path predefined two points. Experiments are performed in the laboratory in which some tables, partition wall, desks, seats and book shelves take part in and the metric grid is 15 cm half the length of the robot. Fig. 7 is the snapshot of the simulator window while building map and Fig. 8 presents the process of getting topological motion command automatically from the topological information as starting points, passage points and target point which made up of three components of coordinate points and direction angle as (x_i, y_i, θ_i)

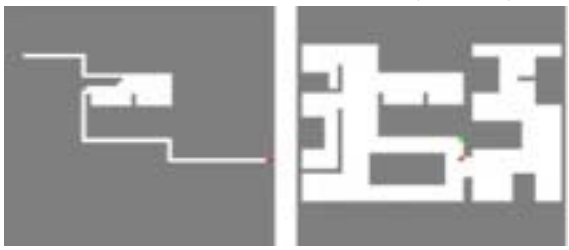

Fig. 7 Map building process on the simulator

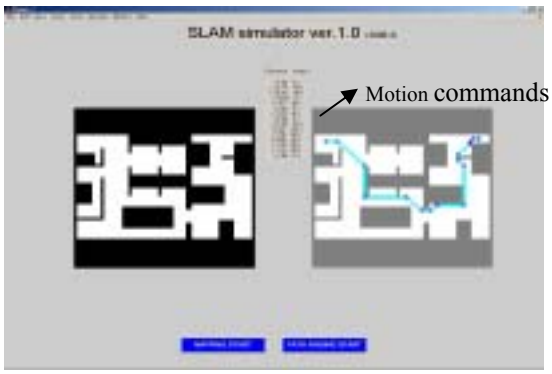

Fig. 8 Path finding process getting from topological information

To verify the location accuracy and control performance of our SLAM system with the other one, we executed following experiments.

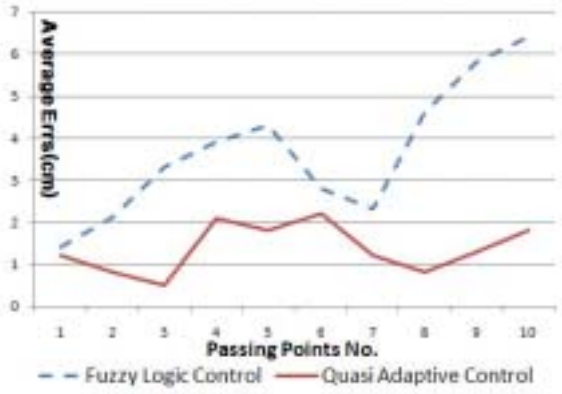

Fig. 9 Comparison result of control performance

As you can see from the above figure, the experiment is about the comparison of the other work with ours i.e. quasi adaptive control about the averaged error result from the starting point to 10 passage points on the robot's trajectory. The averaged error is the sum total of each error's root mean square. Suggested algorithm has the superiority of about 2.5 times to the ordinary fuzzy logic control method implemented in the Begun's work. We think this is because the mobile robot is controlled by independent two wheels and devised algorithm imports each sectional error information and compensation function of exponentially increasing according to absolute error.

VI. Concluding Remarks

To Increase the localization accuracy and automation availability we devised some control algorithms and integrated system with hybrid map. In building a map the robot searches all the environmental area using some coverage pattern algorithm known as backtracking. Maps are metrically formed by the grid cells in the memory database. On the simulator window mapping process is shown in real time. After mapping process, transformation works of the topological information from metric map are performed and from this we can acquire some motion commands specifically how long the robot will go straight ahead and where it will change its direction. In the localization image, optical flow algorithms are implemented using additional image layer taking only part in the localization. PD based adaptive control fitted to the two differential wheels are applied to compensate for the planetary position error. As mobile robot moves by the independent two wheels, special control method using the sectional error information was applied. The experimental results show some satisfactory performance and reliable computation complexities. In the near future we are scheduled to study the applicability to outdoor environment regardless of sloped floor or the rough terrains.

References

[1] Richard Szabo, "Topological navigation of simulated robots using occupancy grid," Int. J. of Advanced Robotic Systems, Vol.1, No.3, 2004.

[2] Ananth Ranganathan, "Bayesian inference in the space of topological maps," IEEE Trans. on Robotics, Vol.22, No.1, 2006.

[3] Shuzhi Sam Ge, Frank L.Lewis, "Autonomous mobile robots: sensing, control, decision making and applications," CRC Press, 2006.

[4] Jose A.Castellanos, Juan D.Tardos, "Mobile robot localization and map building: A multisensory fusion approach," Kluwer academic publishers, 1999.

[5] Michael Montemerlo, Sebastian Thrun, "FastSLAM: A scalable method for the simultaneous localization and mapping problem in robotics," Springer, 2006.

[6] A. Elfes "Using Occupancy Grid" *IEEE Journal of Computer,* Vol. 22, No. 6 pp 46-57, 1989.

[7] B.Kuipers, Y.T.Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *Journal of Robotics and Autonomous Systems*, 1991.

[8] G. Weiss, E.Puttkamer, "A map based on laser scans without geometric interpretation," Intelligent Autonomous Systems, pp. 403-407, IOS Press, 1995.

[9] Tae-Bum Kwon, Jae-Bok Song, "Thinning-based topological map building for local and global environments," *Journal of Control, Automation and Systems Engineering*, Vol. 12, No.7, July 2006.

[10] M.Begun, G.K.I. Mann, R.G.Gosine, "Integrated fuzzy logic and genetic algorithmic approach for simultaneous localization and mapping of mobile robots," *Applied Soft Computing*, Vol.8, pp. 150-165, 2008.

[11] M.Skubic, G.Chronis, P.Matsakis, J.M.Keller, "Generating linguistic spatial descriptions from sonar reading using the histogram of forces," *Proc. 2001 IEEE Int. Conf. on Robotics and Automations*, Seoul, Korea, 2001.

[12] G. Weiss, E.Puttkamer, "A map based on laser scans without geometric interpretation," Intelligent Autonomous Systems, pp. 403-407, IOS Press, 1995.

[13] C. C. Wang and C. Thorpe. "Simultaneous localization and mapping with detection and tracking of moving objects," *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[14] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. "Estimating the absolute position of a mobile robot using position probability grids," *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1996.

[15] J. J. Leonard and H. F. Durrant-Whyte, "Directed sonar sensing for Mobile Robot Navigation," Kluwer Academic Publishers, Boston, MA 1992.

[16] Thomas Röfer, "Using histogram correlation to create consistent laser scan maps," *Proc. of the IEEE International Conference on Robotics Systems(IROS)*, EPFL, Lausanne, Switzerland, pp. 625-630, 2002.

[17] Agostino Martinelli et al, "A relative map approach to SLAM based on shift and rotation invariants," *Robotics and Autonomous Systems*, Vol.55, pp.50-61, 2007.

[18] C. C. Wang and C. Thorpe. "Simultaneous localization and mapping with detection and tracking of moving objects," *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

Suhyeon Kim lives in Daejeon, Korea and was born on December 18, 1975. He majored the mechanical engineering in Korea University in Seoul, Korea and graduated in 2001. He was granted a Master's degree on that field in same graduate school in 2003 and mastered the doctor's course at KAIST in 2005. He is now in the graduate school of Mokwon University. for doctoral degree.

He had spent his youth for the military service in Korean Army for about two years. He has been working as a Public servant in the Daejeon Metropolitan City in Korea since 2007.

Mr. Kim has special interest in the field of humanoid robot and its intelligence. On that field he is studying on the humanoid recognition and inference with stereo vision and the programmatic automaticity.

Hyungrae Kim lives in Daejeon, Korea and was born on January 21, 1988 in his hometown Seoul, Korea. He majors in the intelligent robot engineering in Mokwon University and he is now studying as a junior. He had been trained in special class during his middle school ages and he has the special programming technique using various types of software such as Visual Studio for visual programming, IAR embedded workbench for hardware ROM programming and so on.

He has a lot of knowledge and experience on the digital image processing. And by his good achievements he received the scholarship in the last semester.

Mr. Kim has special interest in the field of mobile robot and its mapping process. He is developing the improved SLAM simulator to the previous version.

Tae-Kyu Yang lives in Daejeon, Korea and was born on February 7, 1958. He majored in the electronic engineering in Gwang-woon University in Seoul, Korea and graduated in 1982. He was granted a Master's degree in 1984 and a doctoral degree in 1989 at the same graduate school. He is now a professor in the department of intelligent robot engineering in Mokwon University.

He had been a full-time lecturer for two years since 1991, had worked as an assistant professor for 4 years since 1993 and then had worked as an associate professor for 5 years since 2002. He had the experience of consecutive filling various posts in the university, for example the Dean in the Student bureau and in the electronic department. He wrote eight books in his major, about 90 technical papers and finished successfully more than 40 projects on the research and development.

Mr. Kim is doing his best in increasing the quality and practical integration about robot education. Also he has very active attitude on the fusion of some other fields or faculties for new technical development.