The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Autonomous Generation of Behavioral Trace Maps Using Rescue Robots

Hayato Mano, Katsunori Miyazawa, Ranajit Chatterjee and Fumitoshi Matsuno

***Abstract*— In the current research we consider the scenario of post disaster rescue operations using multiple robots inside damaged buildings where the damage is not very extreme. The objective the of the work is to develop systems to support efficient rescue missions involving creation and exchange of environment maps enriched with information important to rescue operations, among multiple robots and rescue workers. In current days many research institutes are developing robots with widely different types and capabilities to support rescue operations. In rescue missions involving multiple robots it is essential to have sharable map information for efficient search and navigation irrespective of the differences in the structures and capabilities of the participating robots. In this paper we address the above issue by proposing a map description which combines the geographical data as well as the information about robot behaviors called "Behavioral Trace Map". We describe the algorithm for generation of such maps and we present experimental results using rescue robots developed in our laboratory.**

I. INTRODUCTION

In the recent years various types of rescue robot systems are being developed to work in disaster sites and dangerous environments instead of human workers [1][2]. It is getting necessary to develop systems which can collaboratively support rescue missions involving rescue workers as well as multiple heterogeneous rescue robots [3]. In this context, our project considers the search and information collection task using multiple robots inside damaged buildings. The objective of the research is to generate extended map information on disaster sites and sharing the information among the rescue mission participants. Till date many approaches were proposed regarding generating indoor maps using mobile robots [4]. Research about generating detailed 3D map of the environment using mobile robots are also being considered by various research groups. However, the rescue operation situation often has limitation on time and the amount of data that can be quickly and effectively exchanged and processed by all the participating members. In such situations an easy to exchange topological map of the environment, involving essential information, would be the appropriate approach to meet the requirements [5]. Here the topological map is the environment description with qualitative information about the environmental features like a crossing or a dead-end, The research reported by Nakano et al [5] considers the topological map generation by autonomous mobile robots over a flat surface. However, the real rescue site is expected to be cluttered with obstacles and rubbles, and the robot is expected to face an uneven rough surface. Thus information about the situation of the robot paths and locations of the obstacles are desirable for on-site planning of an efficient rescue mission. In disaster environment, the robot locomotion surfaces usually have features like unstable rubbles, dust covered regions, slippery regions etc. which can not be sensed without actually visiting or moving through those regions. It is also necessary to associate such surface features with the links and nodes of the topological maps while creating the map database. As rescue robot related developments are being advanced by different laboratories all over the world, it may not be ensured that all the participating robots in a rescue mission will have the same mechanisms, locomotion modes or capabilities. To improve the efficiency of a search operation it would be necessary that the optimal robot mechanism is selected based on the information embedded within the map. Thus the map information should be usable by different robot mechanisms independent of their shapes, sizes and locomotion abilities. Further, in the context of any rescue mission, the specific objectives may vary like finding a human victim or locating a harmful object or inspecting the degree of damage in a structure. Thus the generic map information database structure is desired to have extensibility characteristics to easily incorporate new information that are necessary for the specific need of a mission. In this paper we propose a Behavior Trace Map (BTM) which has the following features.

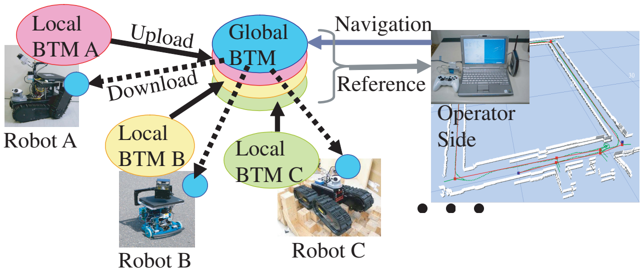


Fig. 1. Framework of Our Project

- It includes obstacle and surface features
- It is usable by different robots having different designs and locomotion capabilities
- It is extensible to incorporate new information

We further propose an approach for automatic generation of BTMs by autonomous mobile robots. Regarding representational and computational overhead, unlike data representation of SLAM based mapping approaches, BTM uses a low memory consuming representation which allows creation and

---

This work was not supported by any organization

H. Mano, K. Miyawawa, R. Chatterjee, F.matsuno are with The University of Electro-Communications, 1-5-1 Chufugaoka, Chofu, Tokyo, 182-8585, Japan `mano, k.miyazawa, ranajit, matsuno@hi.mce.uec.ac.jp`

978-1-4244-3804-4/09/$25.00 ©2009 IEEE

TABLE I

NODE TABLE

NodeID
Coordinate[mm]
Number of links
List of angles between adjuscent links[deg]
Type of Node
Obstacle Shape (if applicable)
Inclination of Slope[deg] (if applicable)
Height of Step[mm](if applicable)
Maximum postural inclination angle of the robot[deg] (if applicable)
. . .

TABLE II

LINK TABLE

LinkID
Start nodeID
End nodeID
Status (Sarched or Un-Searched)
Length of the link[mm]
Minimum path width along the link[mm]
Maximum postural inclination angle of the robot angle[deg] along the link
. . .

usage of the map by robots even with relatively lower end hardware and processing capabilities.

II. INPLEMENTATION ON AN EXAMPLE SYSTEM

In this project we assume the following scenario. Multiple robots are moving autonomously and sensing the surrounding environment using the information from the sensors installed on them. The information collected from the environment are saved in the map database as they are expected to be useful for rescue workers and other robots which may need to move or do other activities in the same environment later. The local BTMs created by the different robots are combined to have a global BTM at a central operation station. In the operation station the behavior trace map is displayed on a GUI. The rescue team is able to identify the geographical or structural shape of the site as well as the location of victims. Also the GUI can be used to instruct the robots to move to any desired location. All the robots can share and use the BTM created by any other robot. Based on the BTM, region status information like already searched regions or accessible regions can be taken into consideration for strategically planning a rescue operation. The Fig.1 shows the framework of the final system of our project. In Fig.1 the firm arrows show the segments where the implementation of the system is completed. The dotted arrows show the portions where the implementation are still in progress. In the following sections we describe the aspects of the system which are already implemented.

A. Components of the BTM

The basic environmental features of the target environment are assumed to be the Passage ways, T junctions, Corners, Dead-ends and Free Spaces. These are the considered to the basic flat surface features. Slope and Stairs are considered to be the basic three dimensional features. The behavioral trace map is described as a collection of Nodes describing the locations having special featurs within the environment and Links interconnecting these Nodes. The Table 1 shows some of the parameters which are registered as node information, and the Table 2 lists a selection of parameters that are registered as link information. The Nodes are considered to be one of the following four types,

- Terminal (Start/End) Node
- Divergence Node
- Free space Node
- Obstacle Node

The Terminal (Start/End) Nodes are created at the beginning or at the end of a search session. The Divergence nodes are created at the Dead-end, Corner, T-joint or Crossing feature locations. The Free-space nodes are created at the wide room like or open space. Obstacle nodes are created at the locations where three dimensional obstacles are detected. At the Obstacle nodes the types of the obstacles are defined in the variable called Obstacle Shape in the data structure. If the obstacle is a slope then the maximum inclination is saved in the data base. In the case of a Staircase type obstacle, the maximum height of the stairs is saved. While running autonomously, if a robot encounters environmental situation like a damaged pathway or a slippery floor or if the autonomous navigation become confusing for any reason, the robot may await intervention form the operator. The situation, under which the operator intervention was required, can also be saved as part of the node feature to register all salient situations encountered by the robot. The Link table registers the length and the minimum width of a path way. A parameter called Status is assigned with each link which indicates if the link is already explored or not. The unsearched status of a link implies that the link is connected to a node only at one of the ends and the environment at the other end is still unknown. While creating a node or a link, a ID number is automatically assigned. The method of creating the nodes and the links are discussed in section III.

B. Use of BTM by different robots

The BTM takes into consideration that the BTM user robot is expected to be different than the one which created the map. When the BTM creator robot encounters any slope or stair the parameters indicating the difficulty level of the obstacle (like maximum slope or maximum stair height) are saved in the BTM. When a different robot is using the BTM it can check the difficulty level of a path and compare that with its own capability to plan the search strategy. If the BTM user robot finds that the minimum path width is less than its own width, then it may decide to take a different route ahead of time.

C. Extensibility of BTM

In the earlier section we mentioned about 4 basic types of nodes but depending on the availability of specials sensor a robot may be capable of identifying new features in the environment. The BTM data structure is designed to incorporate the new features in the database as they become necessary. We have developed infrared sensor unit[6] which a robot can use to locate a victim by finding the distance and angle of a heat source. For example, if a robot carrying this unit is used to create a behavior trace map, it would be possible to include victim location in the BTM.

III. AUTONOMOUS GENERATION OF THE BTM

A. Node

1) Divergence Node: Using the range data from the laser range finder (LRF), mounted horizontally on the robot, the

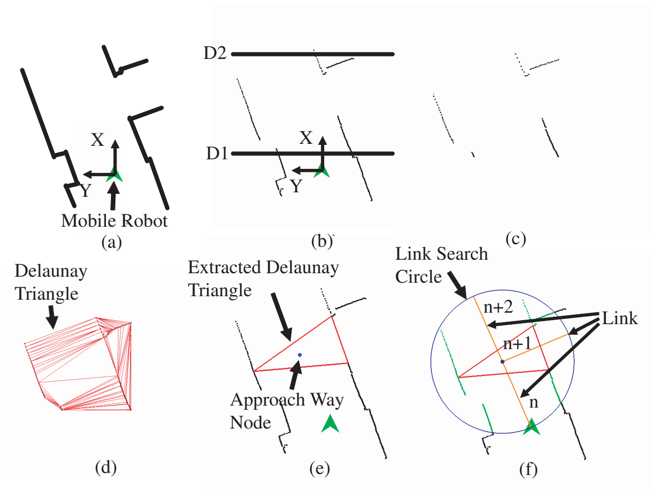


Fig. 2. (a)-(d): Creation of Divergence Node, (f): Creation of Link

Delaunay triangle partitioning is applied and Dead End, Corner, Divergence (T-Joints, Cross Joint, etc.) are recognized and a Divergence joint is created [7]. As shown in Fig.2(a), depending on the geometrical features of the environment, from the acquired data at any point of time, the point data belonging to the range, $\Omega_1 = \{(x_j, y_j)|D1 < x_j < D2\}$, shown in Fig.2(b), is selected to have the Delaunay triangle partitioning (Fig.2(c)). In the current work, among the created Delaunay triangles Fig.2(d), any triangle having obtuse angle or triangles with any side less than the width of the robot, are discarded. If more than one triangle is left, the one nearest to the robot is selected. Subsequently, the circum center of the triangle is found as shown in Fig.2(e). Free Space Nodes are created where at least one side of the robot is free from any obstacles within the selected visibility range of the robot. As shown in Fig.3, along the positive X axis direction with respect to the robot coordinate at a distance of L, a region of a fixed width (the gray portion in the map) is defined, and the minimum distance along the Y axis upto the left wall is defined as L_r and that upto the right wall is defined as L_l. This approach of minimum value computation over a broad region reduces the possibility of generating redundant Free Space Nodes. Unless a candidate free space is wider than the width of the gray area, the feature is not considered to be a Free Space Node. Next, in the environment where, $L_l > C_1$ or $L_r > C_1$or both $L_l, L_r > C_1$, the environmental feature is defined as a free-space where C_1 is a constant (the maximum sensing distance of the LRF). However, if there exist a free space, the robot would create two free space nodes, one at the entrance and the other at the exit of the region. In the situation shown in Fig.3, where $L_l < C1$ and $L_r > C1$, it is concluded that there is a free space on the right side of the robot and a Free Space Node is created.

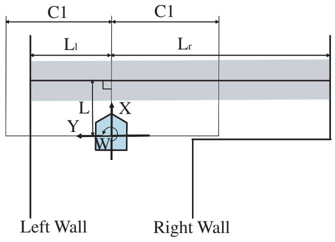


Fig. 3. Free Space Node

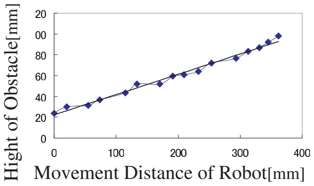


Fig. 4. Obstacle profile of a slope, as perceived from the Laser Range data

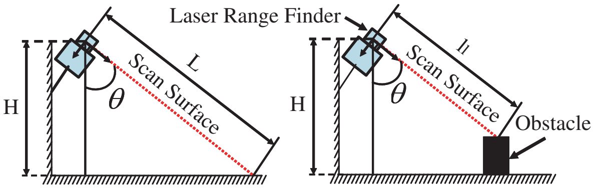


Fig. 5. Obstacle Node

2) Obstacle Node: The robot can sense the slopes, stairs or other three dimensional feature, using data from a LRF unit attached on the robot body at an angle facing downwards (Fig.5). In the case when no obstacle is present, the shortest length of scan range in any direction along the scan surface is considered to be L, the sensor height be H, and the attachment angle is θ (Fig.5). When an obstacles is present, the shortest length of scan range in that direction along the surface upto the newly perceived obstacle is defined to be, l_1 and the height of the obstacle can be estimated as $(L - l_1)\cos\theta$. The length of the obstacle in the direction of motion of the robot can be estimated from the odometric data as the robot moves across the obstacle. As shown in Fig.4, the obstacle height (y axis) is plotted against the obstacle length (x axis) and the variance of the least square fit line throught those points is refered as V. If V is more than a threshold V_{max}, it is decided to be a slope type obstacle. If the V is more than V_{max}, a stair type obstacle. As an example, if the above mention representation is considered for a slope, we get a graph as shown in Fig.4. The straight line on the graph is least squares approximation line. The algorithm used to autonomously generate a link is discussed below. As shown in Fig.2(f), a circle of appropriate radius encompassing the triangle is drawn, which is considered to be the search circle for the links. The point data within this search circle in polar form with respect to the robot coordinates are represented as, $\Omega_2 = \{(r_i, \theta_i)|1 \le i \le N\}$, where, N is the number of points located within the search circle. A data sequence $\{P_n\}$ is defined where the elements of Ω_2 are sequenced such that the condition $\theta_i < \theta_{i+1}$ (for i = 1 to N) is satisfied. Now, if the distance between two adjacent points P_i and P_{i+1} is more than the width of the robot, the opening is considered to be large enough through which the robot may pass. Subsequently a line joining the circum center of the triangle with the mid point of those two adjacent points is selected to be as a new Link originating from that Node.

IV. AUTONOMOUS LOCOMOTION

The autonomous locomotion algorithm is implemented for the locomotion of the Behavior map creator as well as for the map user robots. The implemented locomotion behaviors can be broadly classified as A) Basic locomotion strategy and B) Path feature dependent locomotions.

A. Basic locomotion strategy

The basic locomotion strategy consist of target approach behavior and wall following behavior. An example of the robot behavior while moving towards the target location is shown in Fig.6. Here the translational velocity v and the rotational velocity ω are decided as follows.

$$v = V_c \tag{1}$$

$$\omega = k_w \alpha \tag{2}$$

Where, V_c is the specified approach velocity, k_w is the rotational velocity coefficient and α is the angle of the target point with respect to the robot coordinate. Again, if the target point is expressed in terms of the robot coordinate (L, S), the rotational velocity becomes as follows.

$$\omega = k_w \tan^{-1}\left(\frac{S}{L}\right) \tag{3}$$

Depending on the locomotion state, the robot is controlled by assigning new target points (L, S). In the case of existence of obstacles along the path of the robot, the horizontal LRF data is used to avoid the obstacle. For example, as shown in Fig.7, in the absence of any obstacle the robot uses the target approaching behavior (shown by the farm line arrow) to move towards the goal. If the robot encounters any obstructions and can not reach the target point, the wall following mode is used (as shown by the dotted line arrow). Once the obstacle is avoided, the target approach behavior is re-invoked to approach the target position, using equations (1) and (3).

B. Path feature dependant locomotion

Fig.8 shows the flow-chart for selection of the appropriate autonomous locomotion modes from the implemented options. We explain each of the locomotion modes below.

- Dead End Run Mode

Similar to the section III.A.2, we define the variables L_l and L_r. But in this case, for estimating the environmental situation immediately surrounding the robot as well as that at a distance in the direction of current heading, a portion of the LRF range data is used which fall within the gray area shown in Fig.9. This way the robot can take early decisions related to locomotion strategy from a distance as well as be aware of the possibilities of collisions with immediately near obstacles or other environmental features. If during a run, the $L_l + L_r < C_2$ (where, C_2 is the width of the robot) is satisfied, it is concluded that the path (node) feature to be a dead-end, and it performs a on-the-spot turning operation until $L_l + L_r > C_2$ condition is satisfied.

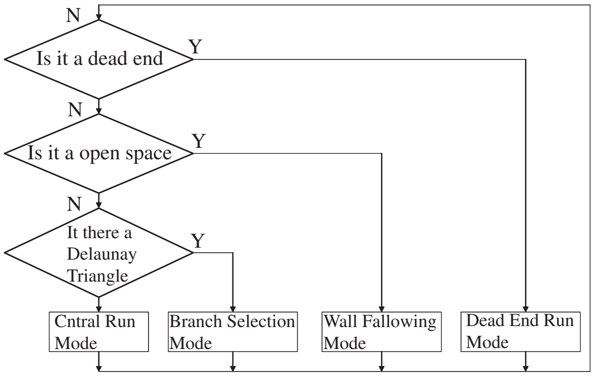


Fig. 8. Flow Diagram for Run Mode selection

- Wall Fallowing Mode

Similar to the definition of Free space, in section III.A.2, if the L_l or the L_r is less than a certain distance, the environmental feature (Node) is considered to be a free space and, similar to the obstacle avoidance mentioned earlier (section V.A), a wall following behavior is performed. For $L_l > L_r$ the robot follows right wall and for $L_l < L_r$ it follows the left wall. In the rare case that both sides are found to have free space at the same computation cycle, then the side with the nearest obstacle in the last computation step is selected for wall following.

- Branch Selection Mode

If a Delaunay triangle is identified during exploration, it is perceived to be either a divergence point or a dead-end. First, as shown in Fig.10, the robot uses the goal seeking algorithm to approach the circum center of the Delaunay triangle as the next target point. When the robot reaches within an area with a radius C_3 around the circum center, a target point is set, on one of the un-searched links, at a distance of C_4 from the circum center of the Delaunay triangle, and the robot moves towards that new target.

- Central Run Mode

In situations other than the above cases, the environment is considered to be a passage way. In reference to the equation (3), the S is set to be,

$$S = \frac{L_l - L_r}{2}. \tag{4}$$

This setting allows the application of the target approach algorithm to make the robot move along the center of the passage way (Fig.10).

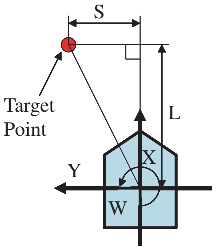


Fig. 6. Target approach behavior

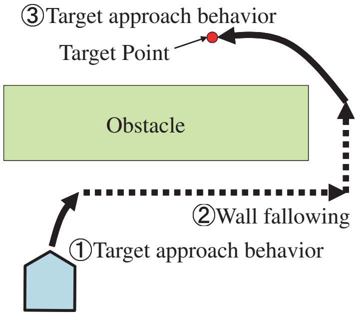


Fig. 7. Obstacle avoidance behavior

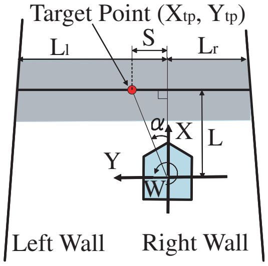


Fig. 9. Situations for Central Run Mode and Dead End Run Mode selection

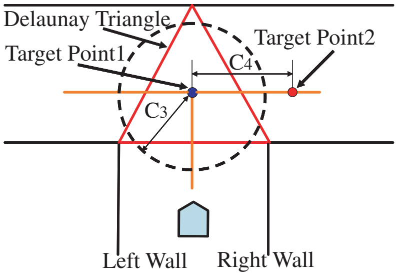


Fig. 10. Branch Selection Mode

TABLE III

SPECIFICATION OF KAMUI

Name	KAMUI
Length	700mm
Width	390mm
Height	740mm
Mass	25kg
Speed	7km/h
Climbable angle	45deg
Communication	wireless LAN
Camera	Wide Angle Camera,WEB
Sensor	Encoder,Laser Range Finder,3D Attitude sensor
PC	Lenovo X61(Robot side) Lenovo X32(Operator side)

V. EXPERIMENT

A. *Experimental Setup*

We tested the effectiveness of the system with the robot called ”KAMUI”(Fig.11) which is an autonomous mobile robot. KAMUI is a crawler type robot with two flipper arms on its sides. Table.III shows the basic specifications of KAMUI. KAMUI acquires obstacle distance information using two Laser Range Finders installed on it. In addition, KAMUI can measure the angle of its posture from a three-dimensional inclination sensor.

Fig.13 is an illustration of system which we used in the experiment. The robot collects sensor data with a laptop PC which is attached with the robot and generates a BTM while doing an autonomous run.

The operator side is equipped with a graphical interface where the operator can observe the map creation process on line. The major information displayed on the operator interface includes the range data acquired from the Laser Range Finder, the Delaunay triangles formation during search of a potential Divergence node, the position of the Target Point, the current Run Mode, the quality of radio communication, etc. An image of the interface is shown in the Fig.12.

The experiment environment consists of situations, as shown in Fig.15, namely, acouple of passages, multi-link junctions, an elevator hall (a potential free space), and a slope. Fig.15 shows a schematic view of the environment used for the experiments. KAMUI ran autonomy in environment composed of the element mentioned above and performed BTM generation automatically. In addition, when a link is generated at a divergence node, the robot moves out of the node along the first link detected in the counterclockwise direction from the entry link around the newly created node.

B. *Experiment Result and Discussion*

KAMUI ran autonomously according to algorithms explained in Section IV, manifesting the following major behavioral steps.

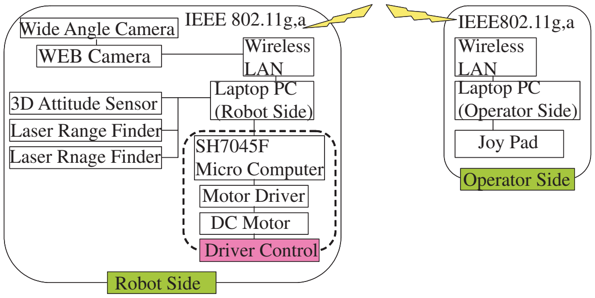


Fig. 13. System Diagram of KAMUI

①KAMUI departed from a start spot and ran along the middle of the passage
②KAMUI ran across a slope of an angle of 10 degrees
③KAMUI recognized a divergence on the path and turned right
④KAMUI recognized free space and ran along the right wall
⑤KAMUI recognized divergence on the path and turned right
⑥KAMUI went to the goal point

During the experiment, KAMUI was able to generate a BTM as shown in Fig.16. The smal squares on the robot trajectory indicate the nodes created during the run. The different types of nodes created by the robot are marked on the image. As discussed in sction 3.A.2, a pair of free space nodes are generated for the free space of the elevator hall, one at the entrance and one at the exit point of the hall. The straight lines, connected to the nodes, indicate the links generated for the BTM. The short line segments which are connected to a node only at one side of the link, indicate the un-searched links. The graphical representation of the BTM consists of these nodes and links only. The environmental artifacts, that are shown around the BTM graph using discrete points, are output of SLAM which we generated by ICP algorithm[8]. And the curved line shows the trace of the robot trajectory. As can be observed in the Fig.16, the slope type obstacle nodes are generated whenever the locomotion surface inclination changes beyond a threshold value. Thus for a triangular type slope three obstacle nodes are created, at the start, at the top and at the end of the triangular obsatacle. The system was able to fill up the node table and the link table automatically as explained in Section III. The robot automatically created a obstacle node when a slope is sensed using the laser range finder. Then it passed over the slope and the maximum inclination angle is saved as a parameter of that obstacle node. The link table was able to appropriately store the narrowest width along the stretch where KAMUI moved along a link. Thus the robot demonstrated the autonomous node and link generation capability. As this automatically generated BTM contains all the necessary information for

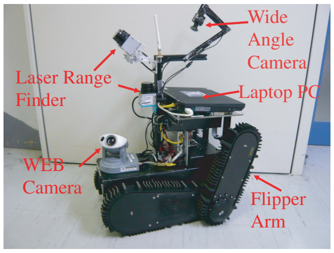
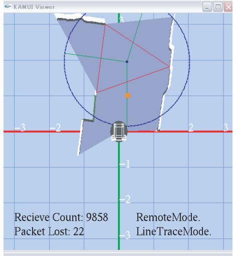


Fig. 11. KAMUI



Recieve Count: 9858
Packet Lost: 22
RemoteMode.
LineTraceMode.

Fig. 12. Interface of BTM system

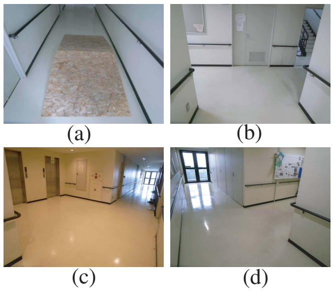


Fig. 14. The real experiment environment

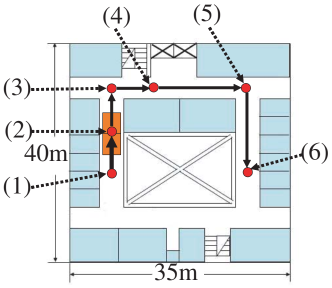


Fig. 15. Schematic view of the experiment environment

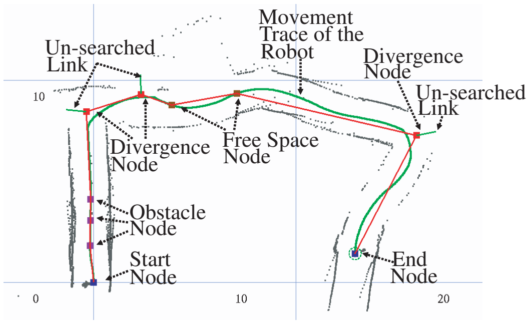


Fig. 16. Experiment Result 1(BTM generation using KAMUI)

rescue strategy planning any other robot or planning system may use this BTM irrespective of the entity which created the map. In addition, we tested the BTM generation run for five times in similar experiment environment and was able to generate exactly the same BTM. It may be said that this system has repeatability that is necessary to have robustness against minor sensor errors or minor changes in the environment. As the movement mechanism of the robot we used, is crawler type, it has inferior Odometry precision. SLAM output gets discontinuous due to accumulation error of Odometry when we performed a wide area search for a extended period of time. However, most of the accumulation errors of Odometry do not influence generation of BTM because BTM uses Odometry only locally. Therefore, it may be said that the BTM is effective to make the map over a wide area. Fig.17 is the BTM that was generated by a wheel type robot. However, as the wheel type robot was not capable to cross over the slope type obstacle used for experimenting with KAMUI, the slope obstacle was removed from the environment during the run. Now, if the Fig.16 and the Fig.17 are visually compared, it can be observed that the robot trajectory as well as the nodes created by the two different robots are similar (except the slop region which was not present while using the wheeled robot). Therefore, it may be said that this system is effective in the sense that the basic BTM created by the robots are mostly independent of the type of the robots. Thus the map generated by one type of robot can be shared and be used by other type of robot platforms.

VI. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

We suggested Behavioral Trace Map (BTM) for use in the rescue scenarios. The details of BTM representation and benefits are presented. The BTM representation allows us to represent map information which is richer in content compared to traditional topological maps but less memory and computation intensive compared to SLAM. The application area is considered to be rescue operations in disaster scenarios. Thus the map information is expected to be usable by different kind of robots having different degrees of locomotion, sensing and map data processing capabilities. We discussed and experimentally demonstrated strategies for autonomous navigation and creation of the BTM using mobile robots.

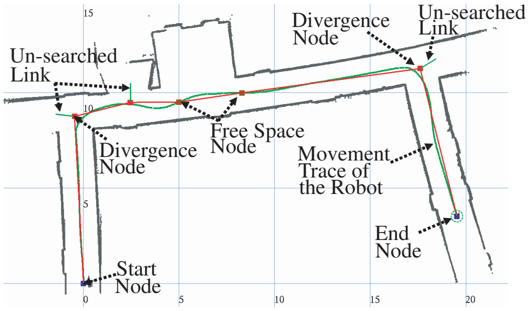


Fig. 17. Experiment result 2(BTM generation using a wheeled robot)

B. Future Works

One of the major issue of topological map generation is the loop-closing problem [9],[10] We are currently implementing and testing some approaches to address that issue. We are also experimenting with autonomous robot navigation using the maps generated using the system reported in this paper. The extensions of the system, to support convenient incorporation of new information in the BTM database, are also being carried out.

The long term goal of this work, which is to build a system for collaborative creation and usage of map information by heterogeneous robot platforms are being pursued. We are addressing the problems of path planning by heterogeneous robots, with different locomotion capabilities and search purposes, using a common BTM representation.

VII. ACKNOWLEDGMENTS

The authors would like to thank to New Energy and In-dustrial Technology Development Organization (NEDO) in the "Project for Strategic Development of Advanced Robotics Elemental Technologies" for supporting our project.

REFERENCES

[1] S.Tadokoro and T.Takamori et al., On robotic rescue facilities for disastrous earthquakes -from the Great Hanshin Awaji (Kobe) Earthquake-, Journal of Robotics and Mechatronics, Vol.9, No.1, pp. 46-56, (1997)

[2] F.Matsuno and S.tadokoro et al., Rescue robots and systems in Japan, Proc. IEEE Int. Conf. on Robotics and Biomimetics, pp. 12-20, (2004)

[3] T.Kamegawa and K.Saikai et al., Developmen of grouped rescue robot platforms for information collection in damaged buildings, Proc. SICE Annual Conference, pp. 1642-1647, (2008)

[4] C.Dornhege and A.Kleiner et al., Behavior maps for online planning of obstacle negotiation and climbing on rough terrain, Proc. IEEE Int. Conf. on Intelligent Robots and Systems, pp. 3005-3011, (2007)

[5] T.Nakano and R.Chatterjee et al., Perception-based Qualitative Map Building using Autonomous Mobile Robots, Proc. IEEE Int. Conf. on Industrial Informatics, pp. 627-632, (2005)

[6] H.Mano and K.Kon et al., Treaded Control System for Rescue Robots in Indoor Environment, Proc. IEEE Int. Conf. on Robotics and Biomimetics, (2008)

[7] D.Silver and D.Ferguson et al., Feature Extraction for Topological Mine Maps, Proc. IEEE Int. Conf. on Intelligent Robots and Systems, pp. 773-779, (2004)

[8] Y.Chen and G.Medioni, Object modeling by registration of multiple range images, Proc. IEEE Int. Conf. on Robotics and Automation, pp. 2724-2729, (1991)

[9] A.Ranganahan and E.Menegatti et al., Bayesian Inference in the Space of Topological Maps, IEEE TRANSACTION ON ROBOTICS, pp. 92-107, (2006)

[10] Jose-Luis Blanco, Juan-Antonio Fernandez-Madrigal, and Javier Gonzalez, "Toward a Unifield Bayesian Approach to Hybrid Metric-Topological SLAM", IEEE Transaction on Robotics, vol.24, no.2, pp.259-270, (2008)