Multi-Robot Exploration for Search and Rescue Missions — A Report of Map Building in RoboCupRescue 2009 —

Keiji Nagatani, Yoshito Okada, Naoki Tokunaga, Kazuya Yoshida, Seiga Kiribayashi, Kazunori Ohno, Eijiro Takeuchi, Satoshi Tadokoro
Tohoku University
6-6-01, Aramaki-Aoba, Aoba-ku, Sendai, Miyagi 980-8579, Japan
keiji@astro.mech.tohoku.ac.jp

Hidehisa Akiyama, Itsuki Noda,
National Institute of Advanced Industrial Science and Technology (AIST)
Tsukuba Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan
noda@aist.go.jp

Tomoaki Yoshida, Eiji Koyanagi
Chiba Institute of Technology
2-17-1, Tsudanuma, Narashino-city 275-0016, Japan
koyanagi@furo.org

Abstract — Mobile robots may be able to aid rescue crews in dangerous environments during search and rescue missions after natural or man-made disasters. In 2006, we began a research project to realize mobile robots that can gather information rapidly at the first stage of a disaster. 3D mapping, which can be an important aid for rescue crews in strategizing rescue missions, is one of our important objectives. Some fundamental elements to enable 3D mapping have been developed. We attended RoboCupRescue 2009 to validate our integrated autonomous 3D mapping system. We demonstrated our mapping system using multiple-robots on the RoboCupRescue field. In this paper, we introduce our mapping system and report the results from the RoboCupRescue competition.

Keywords: *Autonomous exploration, Coverage path planning, Tracked vehicle, 3D mapping*

I. Introduction

A. Background

Mobile robots (called rescue robots) may be able to aid in search and rescue missions instead of rescue crews in dangerous environments such as during natural or man-made disasters like the great Hanshin Earthquake or the sarin gas attack in a Tokyo subway. To promote the development of search and rescue robots, The Ministry of Economy, Trade and Industry of Japan (METI) and New Energy and Industrial Technology Development Organization (NEDO) have set up Project for Strategic Development of Advanced Robotics Elemental Technologies, Area of Special Environment Robots, RT System to Travel within Disaster-affected Buildings [1].

One of the missions for the rescue robots in the research project is to gather information rapidly at the first stage of a disaster in order to eliminate the risk of a possible secondary disaster to human responders. In this case, 3D mapping is one of the important functions that aid rescue crews in strategizing rescue missions. Furthermore, if the rescue robots have the capability to search autonomously in such an environment, the chances of rapid mapping in a large-scale disaster environment are increased. That is because, typically, there are not so many skilled operators who can control teleoperated rescue robots smartly. On the basis of the above situations, we have begun the development of rescue robots for autonomous mapping with other teleoperated rescue robots cooperatively.

(a) Bird's eye view of the target field

(b) Orange area in the target field

Fig. 1. Target field in RoboCupRescue 2009

B. RoboCupRescue

RoboCup was begun as an annual international competition style since 1997 for the promotion of robotic technologies. The ultimate objective was originally set as the development of a robotic football team that beats a human team. Because of RoboCup's concrete objective and competition style, rapid progress in the field of robotic technologies, such as mechanism, control, sensors, and artificial intelligence, is expected.

In 2001, RoboCupRescue league (real robot league and simulation league) was added to RoboCup for boosting the research in the field of search and rescue robotics [2]. This is a competition of how many victims and how accurate maps the robot can obtain within certain time limits.

Basically, the target field is segmented into three areas:

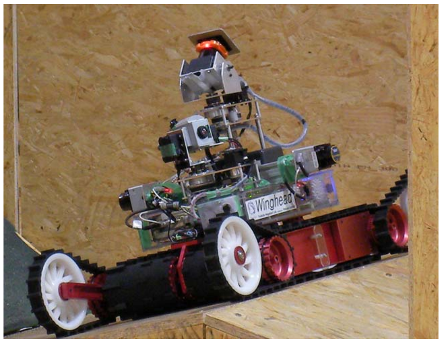


Fig. 2. Tracked vehicle Kenaf 22 with 3D range scanner (at the top)

yellow area, orange area, and red area. The yellow area is designed for autonomous robots. The area contains a path bounded by wooden walls that contains small steps and gentle slopes. The orange area is designed to have the same structure as that of the yellow area, but the slope angles and step heights are relatively larger. The red area is designed for teleoperated robots. The area contains large slopes, stairs, and a rugged terrain. The rugged terrain is made by assembled prismatic vertical wooden poles in different lengths located in a lattice-like arrangement. The rugged terrain is called a “step field.”

In the target field, some simulated victims are embedded to be found by robots. They are characterized by temperature, recorded voices, motions (such as waving hands), CO2 gas emission, and ID tags. Operators should report to referees the above victim information and its location when they find it, and the score is counted by a weighted addition of the information. In practical rescue missions, victim location has to be identified in a map. Therefore, in RoboCupRescue, a higher score is assigned for mapping the victim location.

C. Research objective and issues to be solved

The RoboCupRescue field is, so to speak, a semi toy world. However, if we are not able to perform mapping in such a toy world, mapping in the real disaster environment will be a distant dream. Therefore, we set our first objective as “to map the field in RoboCupRescue.” Furthermore, an additional objective to map the field using multi-robots was set.

To realize the above two objectives, we have to solve the following four issues:

1) Autonomous traversal on an uneven terrain
2) Development of a continuous acquisition system for acquiring 3D environment data
3) Coverage path planning
4) Integration of data obtained by multi-robots

In this paper, we introduce our approach to the above mentioned four issues and report our experimental results of mapping in RoboCupRescue 2009.

II. Autonomous traversal on an uneven terrain

Our objective is to realize autonomous coverage planning and mapping in unknown environments. In the case of a flat surface, conventional wheeled robots are sufficient for this application. However, the target environment shown in section

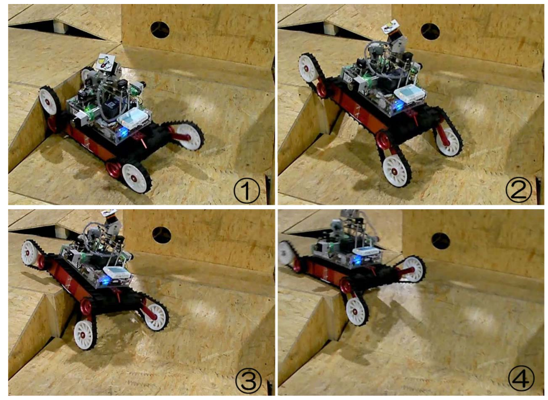


Fig. 3. Maintaining stability by autonomous control of sub-tracks

I-B is imitated as a disaster environment that includes slopes, bumpy surfaces, and step fields. Therefore, an autonomous system for traversal on an uneven terrain is required for our target robot to enable the first proposed issue. For this purpose, some related works were reported (e.g., [3]).

In this competition, we used our tracked vehicle named “Kenaf 22” (Fig. 2) that was equipped with four sub-tracks to maintain the stability of the body with changing mounting angles of sub-tracks actively. Originally, it was designed for teleoperation; however, we applied the autonomous control system of actuated sub-tracks, which was developed by our research group [4].

In the control system, terrain information is obtained by using laser range scanners attached on both sides of Kenaf. The posture data of the body is obtained by a gravity sensor and 3-degrees-of-freedom gyroscopes that are attached to the robot body. The robot controls the motions of sub-tracks to improve the stability of the robot on the basis of the posture data of the body and the terrain data.

In the target environment in RoboCupRescue 2009, the abovementioned system worked pretty well. Basically, the organizer of the competition assumed that the autonomous robots worked only in the yellow field that included small slopes and steps. However, our autonomous traversal system performed autonomous navigation not only in the yellow field but also in the orange field that included large slopes and steps (the maximum gap was approximately about 30cm). Fig. 3 shows that our robot traversed a set of challenging steps autonomously. We attempted the red field that included step fields, but the robot sometimes tipped over. Potentially, the robot could surmount step fields because the teleoperated version did so. Therefore, we understood that local path planning with the consideration of stability was very important for such applications.

III. Continuous acquisition system for acquiring 3D environment data

For mapping in unknown environments, we used a 3D laser range scanner. To enable non-stop navigation with continuous acquisition of 3D environment data, we proposed a continuous acquisition system for all-terrain robots, in the last international workshop on Safety, Security, and Rescue Robotics

(SSRR 2008) [4]. In this method, the coordinate of the robot is perfectly synchronized with the scan time. Therefore, the robot does not have to stop while the range data are being scanned. We have described this method in detail in [4].

In the target environment in the RoboCupRescue 2009, the abovementioned system worked fine. It obtained accurate 3D data on the basis of the odometry. Fig. 4 in section IV shows an example of the mapped 3D data results. Because of the tracks' slippage caused by slopes and bumpy surfaces, long distance navigation included an odometry error that caused a 3D mapping error, which can be observed in the form of some unsteadiness in the data for walls in the figure. However, a short range map is trustable with only odometry information, locally.

IV. Coverage path planning

In the mission of the RoboCupRescue competition, acquisition of victim location is the most important. This implies that not only finding victims but also coverage and map building are required.

To find imitated victims, we mounted an infrared sensor to detect the high-temperature part as the victim location in the field. In this paper, we focus on mapping the environment, so we skip the issue of finding victims.

To realize an autonomous exploration and mapping in unknown environments, we implemented frontier-based coverage path planning on our tracked vehicle. The coverage path planning that we implemented for the competition repeated the following procedure.

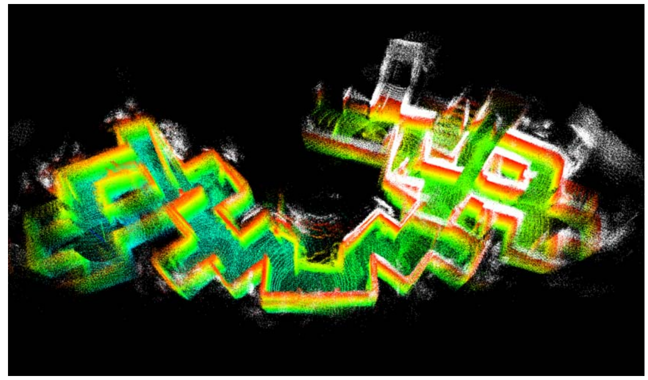
1) Creation of local Digital Elevation Map (DEM)
2) DEM matching to create a large-scale environment
3) Frontier-based coverage path planning

A. Creation of local Digital Elevation Map

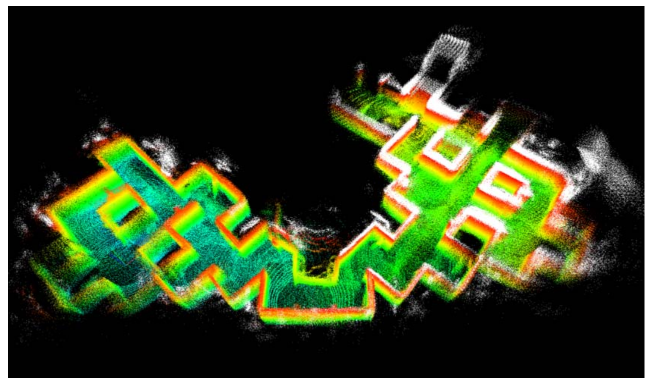
The target robot has the capability to obtain environment data continuously, as discussed in section III. The obtained data is considerably large, so we use the DEM representation for coverage path planning and a position adjustment of the robot [5]. It has a disadvantage that the DEM cannot represent spaces under objects because only the highest scan point is effective. In the RoboCupRescue field, there was no problem because of this disadvantage. A local DEM (5m $\times$ 5m) was generated in every 5s while the robot moved.

B. DEM matching to create a large-scale environment

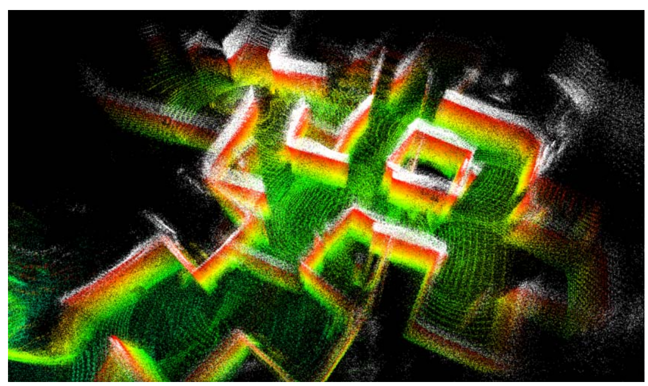
In each creation of a local DEM, we executed scan-matching using the ICP algorithm [6] [7] to correct the unsteadiness of data of walls and to construct a global DEM, online. In this case, we did not consider a loop closing problem. Because there were many environment signatures, the matching did not fail in the target environment. The Fig. 4-(a) shows a representation of point-cloud mapping without ICP, and the Fig. 4-(b) shows the result with ICP. In addition, Figs. 4-(c) and Fig.4-(d) are magnified point-cloud maps of the right parts of Figs. 4-(a) and Fig.4-(b), respectively. It is obvious that the standard ICP algorithm worked well in our case.



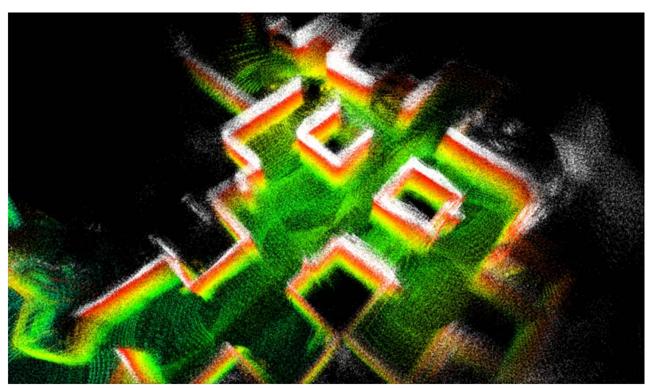
(a) Odometry mapping (3D mapping based on odometry)



(b) ICP mapping (3D mapping based on ICP algorithm)

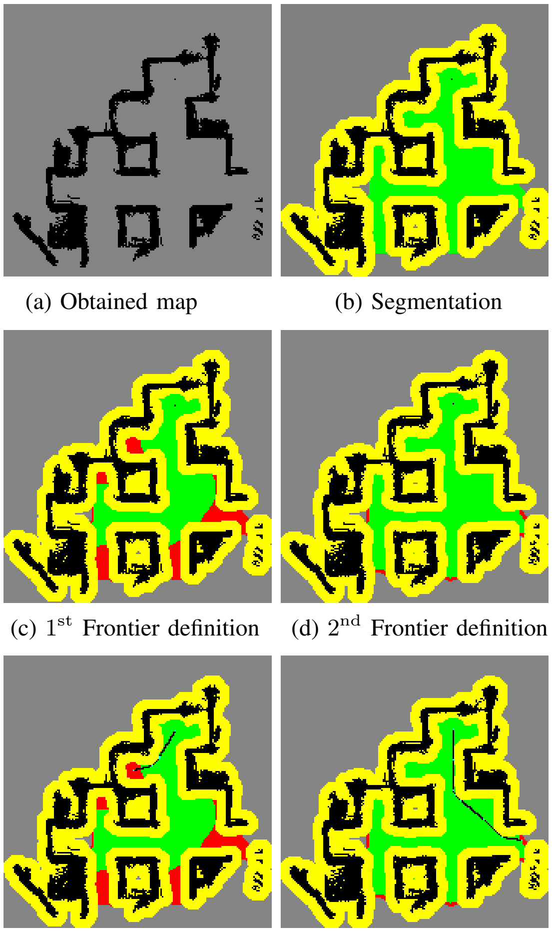


(c) Odometry mapping: magnified in the right part of (a)



(d) ICP mapping: magnified in the right part of (b)

Fig. 4. DEM matching results



(a) Obtained map (b) Segmentation

(c) 1st Frontier definition (d) 2nd Frontier definition

(e) Planned path in (c) (f) Planned path in (d)

Fig. 5. Frontier-based coverage path planning

C. Frontier-based coverage path planning

Using the constructed global DEM, we implemented a frontier-based coverage path planning algorithm [8], as shown in the following procedure.

1) Segmentation of global DEM
An obtained map based on Global DEM is segmented into (1) open space O_s, (2) obstacle space O_b, (3) C-space C_s (which is at a certain distance to obstacles), and (4) unknown space U_s, on the basis of the height data of the DEM. Fig. 5-(a) shows an example of an obtained map based on a global DEM, and Fig. 5-(b) is a segmented result of the map; the green area corresponds to O_s; the black area, to O_b, the yellow area, to C_s; and the gray area, to U_s.
2) Picking up frontier regions
In this research, we set up two definitions of frontier regions F_r that should be explored. The first definition is to consider a distance from the robot's path, defined by the following equation:

$$F_r = O_s \cap C(R_p), \tag{1}$$

where $C(\cdot)$ is the complementary set in the bracket, and R_p is the space where the distance to the vehicular swept path is less than the threshold (in our case, 1.0 m). This means that the frontier regions are assigned as the open places where the robot does not reach within the threshold distance. Fig. 5-(c) shows an example of frontier regions colored by red on the basis of this definition.
The second definition of the frontier regions is to consider the "line of sight" of the laser range sensor. The line of sight is defined as the border of an open space around the unknown space. Fig. 5-(d) shows an example of frontier borders colored by red on the basis of this definition.
3) Path planning and execution
Once the frontier regions are defined, the closest region from the robot is chosen as the sub-goal. To find the shortest path, we apply Dijkstra's algorithm. Finally, a grid-based path is generated. After applying a smoothing algorithm, the robot tracks the generated path to the sub-goal. Fig. 5-(e) shows an example of a planned path to one of the frontier regions based on the first definition. Fig. 5-(f) shows an example of a planned path to one of the frontier borders based on the second definition.
4) Repetition
The algorithm repeats from 1 to 3 until there exist no frontier regions.

In the field in RoboCupRescue, the abovementioned algorithm worked well in terms of exploring the environment. The first frontier definition was applied to the typical competition of searching victims. It took a very long time to explore the target environment. Finally, because of our poor victim search function, the robot did not find any victims in the preliminary stages. However, the algorithm worked well in terms of the coverage of the target environment. The second frontier definition was applied to the autonomous challenge in best in class (an independent competition to validate functions of rescue robots). The mission of the challenge did not require to search victims, instead to obtain the entire environment map. It took approximately only 20 min to map three quarters of the entire environment, as shown in Fig. 4-(b).

V. Integration of data obtained by multi-robots

A. Technical challenge

An exploration in the disaster area using multi-robots is effective for time-constraint searching missions. Therefore, some cooperative mapping algorithms were proposed for such purposes (e.g., [9] and [10]). In RoboCupRescue, cooperative map building using teleoperated robots and autonomous robots is preferred because autonomous robots still find it difficult to enter the orange and red areas in the challenge environment. Therefore, in this competition, we attempted to enable data integration using a database.

To realize the integration of the data obtained by multi-robots, the following issues have to be solved:

- Centralizing map data obtained by multi-robots
- Map data reduction

- Relative position adjustment among robots

In this section, our approach to the abovementioned issues and a performance test are introduced.

B. Centralizing map data obtained by multi-robots

To centralize map data obtained by distributed robots, we used Mitigation Information Sharing Protocol (MISP)[11] as a communication protocol between each robot and our Geography Information System (GIS), DaRuMa (DAtabase for Rescue Utility MAnagement) [12]. MISP is an XML-based protocol. It provides functions to access and to maintain a geometrical information database over networks. DaRuMa is one of the MISP server implementations. DaRuMa works as database middleware and utilizes a MySQL server as its background. Client programs can communicate with DaRuMa using MISP. Further, all registered/queried data are transferred to/from a MySQL server through DaRuMa.

Each robot sends MISP messages that represent either victim data or 3D data with a timestamp and the robot's position to DaRuMa.

C. Map data reduction

Theoretically, each scanned point can be registered as a point object in DaRuMa for searching. Practically, map data obtained by a 3D laser range scanner is so huge that the data has to be reduced. We adopt the DEM representation to register 3D data to GIS for reduction of registration cost and searching cost. Each robot sends local DEM data at a certain interval (every 5s typically).

D. Relative position adjustment among robots

If there are no positioning errors and sensing errors, the obtained map data are integrated in the DaRuMa on the basis of the odometry-based position data. Then, finally, a 2D map is outputted in a geotiff map style. However, there exists a positioning error of each robot. To compensate for this error, we apply a graph-based map correction method. The method is described in detail in [10], and a brief of the method is as follows:

1) Each robot generates DEMs and it defines the position as "nodes."
2) In case the robot recognizes a position relation with error covariance between the nodes by odometry or scan matching, it establishes "constraint relationships" between the nodes.
3) The robot optimizes the nodes' position (trajectory of robots) in order to maximize the joint probabilities of possible constraint relationships.
4) Finally, the robot obtains maps by posting the sensory data based on the optimized trajectory of the robots onto the environment space.

In our implementation, we set two "constraints relationships." One of the constraints is the traveling distance between two local DEMs and its error variance. The other is the ICP matching results [6].

The matching program performs ICP matching of all combinations of obtained DEMs to establish the constraints between nodes. Therefore, the number of combinations is large, and the computing time will be long. To solve these problems, we implemented the matching and optimization processes sequentially. The sequential optimization process optimizes the graph when new DEM data arrives. In the sequential optimization process, the combinations of DEM matching are only between the new DEM and all obtained DEMs, and the result is obtained when the mission is completed.

E. Performance test

We applied the abovementioned method to our test field in Chiba Institute of Technology, Japan. It simulated a standard but relatively small field in RoboCupRescue. We used three robots, called Kenaf-6, Kenaf-21, and Kenaf-22. Kenaf-6 was designed for teleoperation, and it mounted a sensor arm that was capable of searching victims in holes (shown in Fig. 6). Its mapping sensor had to be simple because of space limitations. Therefore, one tilted laser range scanner was mounted at the front of the body to obtain 2D range data. While the robot moved, it integrated the sliced 2D range data for forming 3D range data based on its odometry. Other robots, Kenaf-21 and Kenaf-22, had a 3D laser range scanner.

Fig. 6. Teleoperated robot: Kenaf6

In this implementation, the size of the local DEM was set as 100-grid square, and each grid size was set as 5 cm × 5 cm. One typical result is shown in Fig. 7. In these figures, each L-shaped red mark represents a node that implies the center of the obtained DEM registered discretely. Each green line represents a constraint relationship between the nodes.

Fig. 7-(a) shows the raw data obtained by three robots. Fig. 7-(b) represents a matching result using Kenaf-22's data alone based on our approach. The environment became more accurate, but the loop shown in the circular-part A was not completed because the robot did not explore. Furthermore, the walls were not matched completely, as shown in the circular-part B. On the other hand, Fig. 7-(c) represents a matching result using three robots' data based on our approach. It was observed that the abovementioned problems of using a single robot were solved (in the circular-part A and in the circular-part B) by complemental explorations of other robots. This was a big advantage of the integration of the data obtained by multi-robots.

Based on the abovementioned performance test, we concluded that our approach can be applied to RoboCupRescue missions. In fact, we implemented the above approach, but we did not obtain a good experimental result in the field of RoboCupRescue 2009. The reasons are (1) time constraints of the competition that permits only a small area coverage by autonomous robots and (2) a rare odometry error caused by the angular-speed's exceeding the dynamic range of its gyroscope in case of quick motions of the robots.

VI. Conclustions and future works

In this paper, we reported the mapping result in the field of RoboCupRescue 2009. To implement such map building, we set the four issues to be solved, re-described as follows:

1) Autonomous traversal on an uneven terrain
2) Development of a continuous acquisition system for acquiring 3D environment data
3) Coverage path planning
4) Integration of data obtained by multi-robots

Further, we introduced our approach to solve the abovementioned issues and results. The mapping results in both Fig. 4 and Fig. 7 validated our approach in such a target environment.

In our current implementation, there is no strategy to share the robot's exploration area in our multi-robots exploration. To realize such area planning is one of our future works. Furthermore, the relative position adjustment method shown in section V-D is implemented in ad hoc code. In order to solve these problems, we will extend MISP as a common communication protocol among robots, GIS and matching programs, and will implement their functions in DaRuMa.

Acknowledgment

This work was supported by Strategic Advanced Robot Technology, an R&D project of the NEDO, Japan.

References

[1] Tomoaki Yoshida, Keiji Nagatani, Eiji Koyanagi, Yasushi Hada, Kazunori Ohno, Shoichi Maeyama, Hidehisa Akiyama, Kazuya Yoshida, and Satoshi Tadokoro. Field experiment on multiple mobile robots conducted in an underground mall. In *Field and Service Robotics*, 2009.

[2] Adam Jacoff, Elena Messina, Brian A. Weiss, Satoshi Tadokoro, , and Yuki Nakagawa. Test arenas and performance metrics for urban search and rescue robots. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3396–3403, 2003.

[3] Raymond Sheh, M. Waleed Kadous, Claude Sammut, and Bernhard Hengst. Extracting terrain features from range images for autonomous random stepfield traversal. In *Proc. of the 2007 IEEE International Workshop on Safety, Security and Rescue Robotics*, 2007.

[4] Yoshito Okada, Keiji Nagatani, and Kazuya Yoshida. Semi-autonomous operation of tracked vehicles on rough terrain using autonomous control of active flippers. In *Proc. of IEEE/RSJ International Conference on Intellegent Robots and Systems*, 2009.

[5] E.H.L. Fong, W. Adams, F. L. Crabbe, and A. C. Schultz. Representing a 3-d environment with a 2 1/2-d map structure. In *Proc. of the IEEE/RSJ Conference on Intelligent Robots and Systems*, pages 2986–2991, 2003.

[6] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[7] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proc. of the International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, 2001.

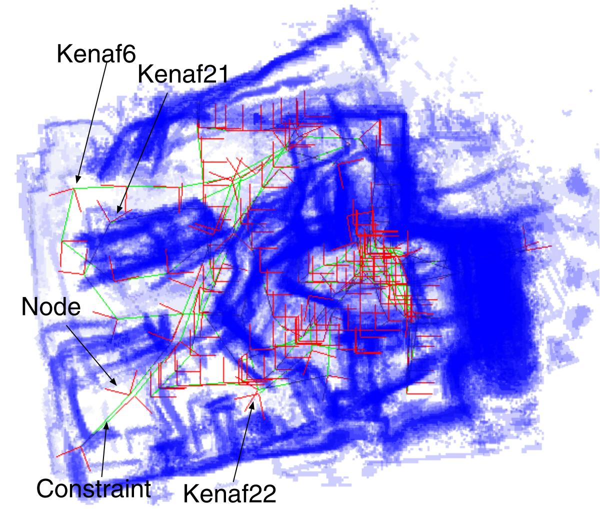
[8] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proc. of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 146–151, 1997.

[9] Max Pfingsthorn and Andreas Birk. Efficiently communicating map updates with the pose graph. In *Proc. of IEEE/RSJ International Conference on Intellegent Robots and Systems*, 2008.

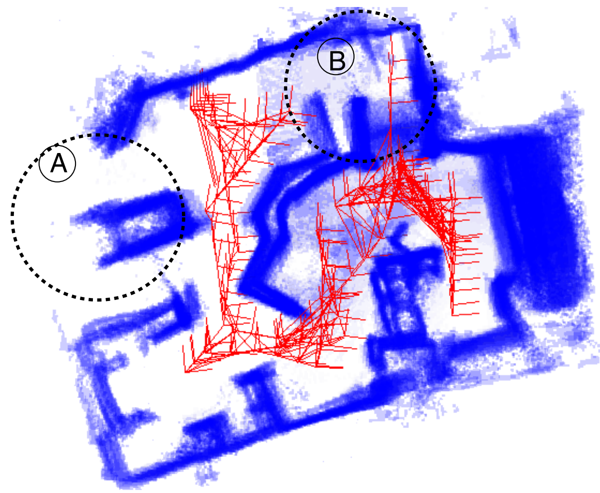
[10] E.Takeuchi and T.Tsubouchi. Multi-sensor map building based on sparse linear equations solver. In *Proc. of IEEE/RSJ International Conference on Intellegent Robots and Systems*, pages 2511–2518, 2008.

[11] NIED and AIST. *Mitigation Information Sharing Protocol*, rev.1.00.028s edition, 2006.

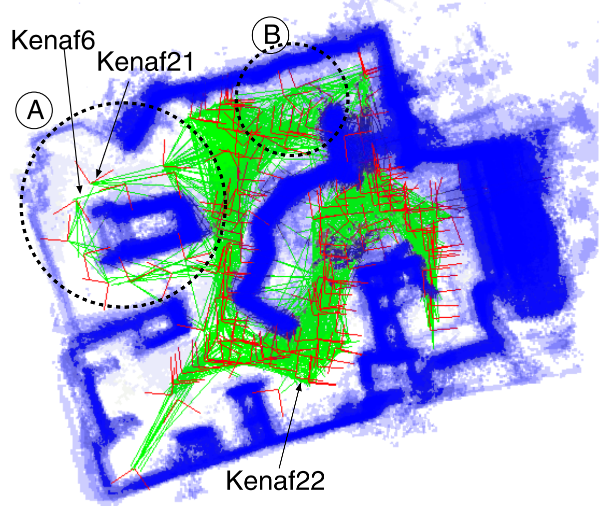
[12] DaRuMa. http://sourceforge.jp/projects/daruma/.



(a) Raw data from three robots



(b) Corrected map using Kenaf-22's data alone



(c) Corrected map using data obtained by all three robots

Fig. 7. Typical result of a performance test