Proceedings of 2013 IEEE
International Conference on Mechatronics and Automation
August 4 - 7, Takamatsu, Japan

Path Planning for Mobile Robot on Rough Terrain based on Sparse Transition Cost Propagation in Extended Elevation Maps

Takeshi Ohki*, Keiji Nagatani*, and Kazuya Yoshida*
*Department of Aerospace Engineering, Tohoku University
Aramaki aza Aoba 6-6-01, Sendai, 980-8579, Japan
Email: {takeshi,keiji,yoshida} @astro.mech.tohoku.ac.jp

***Abstract*— To efficiently evacuate an area threatened by volcanic disaster, onsite observation of active volcanoes by remotely controlled mobile robot systems is desired. Issues involved with developing such systems include planning a safe path for the robots. In this research, our objective was to realize a safe path planning method based on a digital elevation map (DEM) of volcanic mountain fields that considers the mobility of a mobile robot. We assumed that the DEM is obtained through airplane laser measurements beforehand. Because the target environment is vast, obtaining a DEM with sufficiently high resolution is difficult. Even if this is possible, path planning based on such a high-resolution DEM in vast environments significantly increases the computational load. Therefore, we propose a path planning method that can be applied to any DEM resolution; path planning is seamlessly performed roughly in a global scale and precisely in local scales. We extended the general DEM into 3-D space by adding an axis to denote the discrete heading direction of a mobile robot, which we call an extended elevation map (EEM). In the 3-D space EEM, the transition-cost from the start position is derived for each voxel by considering the mobility of the mobile robot. The transition-cost is sparsely propagated from the start position, and the sparsely valued field derives a single path with the lowest transition-cost to reach the goal position. The proposed method was implemented, and simulation experiments using DEMs of real volcanoes were performed to confirm its validity.**

***Index Terms*— Mobile robot, path planning, digital elevation map, volcano exploration**

I. Introduction

A. Background

Since the high-magnitude earthquakes of March 2011, the eruption potential of many active volcanoes in Japan has increased. Once a volcano erupts, it induces severe disasters, such as pyroclastic and debris flows. To efficiently evacuate an area threatened by a volcanic disaster, onsite observation of active volcanoes is desired to generate real-time hazard maps. However, when the eruption alert level is high, approaching such volcanoes is prohibited as human observation is difficult and dangerous.

Based on the above demands, common practices involve remote observations by fixed cameras or unmanned aerial vehicles (UAVs) [1]. However, there observations have limitations. Observation by fixed cameras has a limited field of view. UAV-based observation is limited with regard to the observation time, data resolution, and weather dependency.

Fig. 1. Field Mobile Robot El-Verde in Izu-Oshima island

Therefore, remote-controlled ground mobile robots are desired so that any position of interest can be observed; these robots currently being developed [2]. We have also been developing various types of mobile robot systems for volcanic mountain exploration (Fig. 1) [3][4].

B. Research Purpose

One issue with developing such systems is planning a safe path to the desired observation position. Our objective was to present rough standards for safe paths to the desired position and the required mobility of mobile robots. We considered digital elevation maps (DEMs) obtained by airplane light detection and ranging (LIDAR) sensor; these measurements can be obtained before volcanic eruptions. Airplane-based LIDAR can obtain vast terrain information quickly and efficiently. However, generating a sufficiently high-resolution DEM by remote scanning is difficult. Even if it is possible, path planning based on high-resolution DEMs of vast environments significantly increases the computational load.

In this paper, we propose a path planning method that can be applied to any DEM resolution: path planning is seamlessly performed roughly on a vast scale and precisely at local scale. The proposed method consists of three steps:

978-1-4673-5560-5/13/$31.00 ©2013 IEEE

DEM region and resolution selection, extended elevation map construction, and sparse transition-cost propagation.

In the first step, the target DEM region is selected, and the resolution of the DEM is selected to maintain the voxel total counts. In the second step, the DEM is extended into 3-D space by the addition of an axis to denote the heading direction of the mobile robot; this is called an extended elevation map (EEM), same as [5]. Based on the slope angles of the region occupied by the robot at each voxel in each layer, the posture of the robot can be approximated even in the sparse DEM. In the final step, the transition-cost from the start position is propagated for each voxel in the constructed 3-D space of the EEM based on the transition-cost function while considering the mobility of the mobile robot. To reduce the computational load, the propagation is performed sparsely unlike the general dense propagations reported in [6][7]. If the goal is found, the generated cost field derives a single path from the start to the goal with the lowest cost.

The advantage of this method is adjustment flexibility of the calculation load by several parameters such as the resolution of DEM and heading direction in the EEM, and the sparseness of the propagation. This paper also contributes to show a practical application to a vast real volcanic environment.

In this paper, we introduce the proposed path planning algorithm, its implementation, and several simulation experimental results using the DEM of a real active volcano in Japan.

II. Proposed Method

The proposed method plans a safe path in a vast DEM obtained by airplane LIDAR sensor. The proposed method consists of three steps: DEM region and resolution selection, extended elevation map construction, and sparse transition-cost propagation. In the following subsections, we explain the details of these steps.

A. DEM Region and Resolution Selection

Fig. 2 shows the conceptual diagram of a DEM.

First, the user defines the size of the target area in the obtained DEM. If a user selects the vast region as the target area, the target DEM needs to be culled sparsely to reduce the computational load. In contrast, if the user defines a sufficiently small region, the user simply extracts the desired region without culling.

A cell in the DEM is denoted as $\mathbf{G}(x_n, y_n)$, where x_n and y_n denote the coordinates of the X- and Y-axes, respectively. The cell lengths along the X- and Y-axes are also defined as l_x^{cell} and l_y^{cell}, respectively. The total lengths of the DEM along the X- and Y-axes are defined as l_x^{DEM} and l_y^{DEM}, respectively. The maximum numbers of x_n and y_n affect the computational load and are defined as x_m and y_m, respectively. During the resolution adjustment process, x_m, y_m, l_x^{DEM} and l_y^{DEM} are varied.

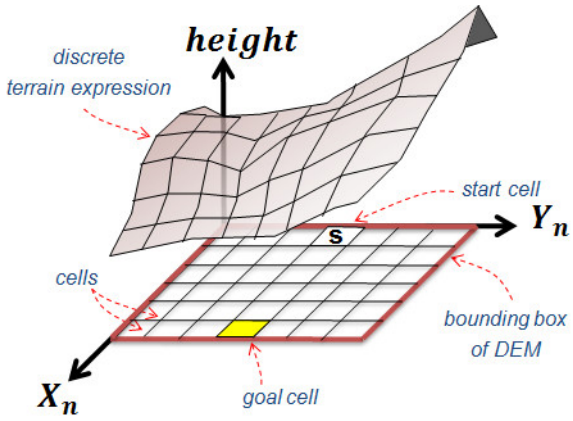

Fig. 2. Conceptual diagram of digital elevation map (DEM)

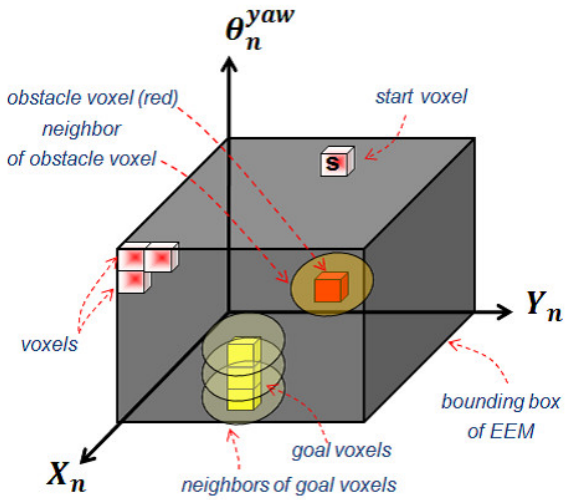

Fig. 3. Definition of extended elevation map (EEM)

B. Extended Elevation Map Construction

The selected DEM is then used to generate an EEM. By adding an axis to denote the discrete heading direction θ^{yaw} of a mobile robot, the DEM is layered to generate the EEM, as shown in Fig. 3. Because the EEM is in 3-D space, $\mathbf{G}(\mathbf{x_n}, \mathbf{y_n})$ in DEM is expressed as voxel $\mathbf{V}(x_n, y_n, \theta_n^{yaw})$ in the EEM, where θ_n^{yaw} is a number that denotes the discrete heading direction of a mobile robot. θ^{yaw} is discretized between 0° and 360° by dividing it with the maximum number of θ_n^{yaw}, defined as θ_m^{yaw}. We call the voxels sharing the same θ_n^{yaw} a $layer$.

key and **val** represent the index and value, respectively, of the voxel contained in the EEM respectively. The pair of **key** and **val** construct the EEM list $\mathbf{List}^{\mathbf{EEM}}(\mathbf{key}^{\mathbf{EEM}}, \mathbf{val}^{\mathbf{EEM}})$. In this case, $\mathbf{V}(x_n, y_n, \theta_n^{yaw})$ is used as $\mathbf{key}^{\mathbf{EEM}}$, and its corresponding value $\mathbf{val}^{\mathbf{EEM}}$ is

TABLE I

DEFINITION OF VOXEL STATE

state S	state meanings	color
$init$	initial state	N/A
$start$	contains start position	white
$goal$	contains goal position	white
$goal_{naer}$	neighbor of $goal$ voxel	yellow
$goal_{reach}$	goal voxel with lowest transition-cost	sky blue
$obst$	prohibited posture	red
$obst_{naer}$	neighbor of $obst$ voxel	orange
reg	voxel storing transition-cost	pink
reg_{near}	neighbor of reg voxel	gray
$reg_{finished}$	voxel that has already finished extending all the paths sparsely	dark red

defined as follows:

$$\mathbf{val}^{\mathbf{EEM}} = \begin{pmatrix} posX \\ posY \\ height \\ \theta^{yaw} \\ \theta^{roll} \\ \theta^{pitch} \\ cost_{trans} \\ S \\ \mathbf{key_{parent}} \end{pmatrix} \quad (1)$$

where $posX$, $posY$, and $height$ denote the planar position and the height defined by $\mathbf{G}(\mathbf{x_n}, \mathbf{y_n})$ in the DEM, respectively; θ^{yaw} is the non-discrete heading angle at layer θ_n^{yaw}; and θ^{roll} and θ^{pitch} are the rolling and pitching angles of the robot when the robot is located at $\mathbf{V}(\mathbf{x_n}, \mathbf{y_n}, \theta_\mathbf{n}^{\mathbf{yaw}})$.

In addition, $cost_{trans}$ is the cost to reach $\mathbf{V}(x_n, y_n, \theta_n^{yaw})$. The state of the voxel is expressed as S. Table I shows the list of S. $\mathbf{key_{parent}}$ represents the key of the connected voxel that generates this voxel and is described later.

$posX$, $posY$, $height$, θ^{yaw}, θ^{roll} and θ^{pitch} are derived as the first step to fill the value in $\mathbf{val}^{\mathbf{EEM}}$. We define the voxel region occupied by a robot as the "robot-occupied region."Based on the average slope angles of the robot-occupied region, the approximated θ^{roll} and θ^{pitch} can be derived at each voxel in each layer, even in the sparse DEM.

Next, if θ^{roll} and θ^{pitch} are larger than the maximum rolling and pitching angles of the robot θ_{max}^{roll} and θ_{max}^{pitch}, the state of the voxel is set as $obst$. The states of the neighbor voxels within radius R_{obst} of the $obst$ voxel are subsequently overwritten to $obst_{near}$ to expand the obstacles not to consider the size of the robot.

The user-defined start and goal voxels are then set, and the voxel states are assigned as $start$ and $goal$, respectively. The states of the neighbor voxels within radius R_{goal} of the $goal$ voxel are also overwritten to $goal_{near}$, but only if the state of the voxel is $init$. Furthermore, if the user does not want to define the heading angle at the goal voxel, the algorithm sets multiple goal voxels in all layers of the EEM.

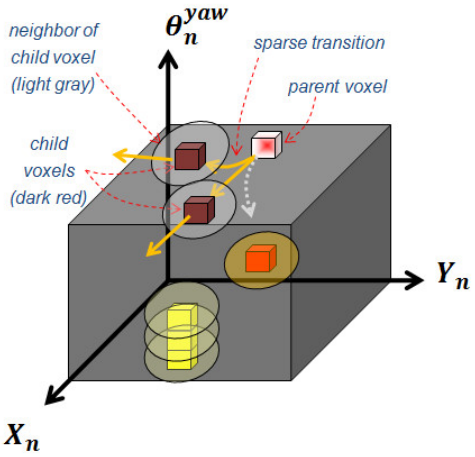

Fig. 4. Sparse transition-cost propagation in extended elevation map (EEM)

C. Sparse Transition-Cost Propagation

As the final step, the algorithm sparsely propagates the transition-costs in the EEM from the start position based on the transition-cost function, which depends on the mobility of the mobile robot, and a single path is derived by extraction of the connected **keys**. This step consists of two parts: sparse transition-cost propagation and path extraction.

The first part involves propagating the cost from a parent voxel to a child voxel, as shown in Fig. 4. The parent voxel starts transiting to several reachable child voxels and overwrites their states to reg only if S of the child voxel is $init$; this is to limit the transition-cost being registered for a voxel only once. To judge whether the child voxel can be reached from the parent voxel, we assume a minimum turning radius of the robot $R_{minturn}$. The distance of the single transition is called segment length l_{seg} which controls the curvature resolution of the generated path. To reduce the computational load, the neighbor voxels within radius R_{reg} of the child voxels are also overwritten to reg_{near}, but only if the state of the child voxel is $init$. Because of the above algorithm, the cost registration is propagated sparsely.

After finishing the transition to all reachable layers, the state of the parent voxel is changed to $reg_{finished}$ if the parent voxel state is not $start$. The all child voxels register their $\mathbf{key_{parent}}$ as the **key** of the parent voxel. $cost_{trans}$ of the child and parent voxels are respectively called $cost_{trans}(child)$ and $cost_{trans}(parent)$, and have the following relation:

$$cost_{trans}(child) = cost_{trans}(parent) + cost_{trans} \quad (2)$$

The transition-cost $cost_{trans}$ is defined as follows:

$$cost_{trans} = cost_{dist} + cost_{turn} + cost_{pitch} + cost_{roll} \quad (3)$$

$$= k_{dist} l_{seg} + k_{turn} \Delta\theta^{yaw} + k_{pitch} \theta_{ave}^{pitch} + k_{roll} \theta_{ave}^{roll} \quad (4)$$

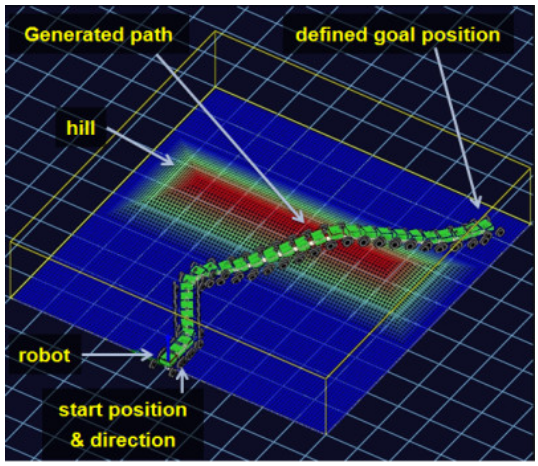

Fig. 5. Found path in simulated environment

where k_{dist}, k_{turn}, k_{pitch}, and k_{roll} are the coefficients of the distance cost $cost_{dist}$, turning cost $cost_{turn}$, pitching cost $cost_{pitch}$, and rolling cost $cost_{roll}$. $\Delta\theta^{yaw}$ is the angle displacement of the heading angle from the child voxel to the parent voxel. θ^{pitch}_{ave} and θ^{roll}_{ave} are the average θ^{pitch} and θ^{roll} of the child and parent voxels, respectively.

Initially, $cost_{trans}$ of the start voxel is set to zero. The above transition-cost propagation is performed from the start voxel as the first propagation. Then, the propagation is repeatedly performed in turns in ascending order from the lowest transition-cost for all voxels whose state is reg. If the propagated child voxel reaches the voxel whose state is $goal$ or $goal_{near}$, the state is overwritten to $goal_{reach}$.

After the propagation procedure is finished, the second part is begun: the path is extracted by tracking back to $\mathbf{key}_{\mathbf{parent}}$ from the voxel with the $goal_{reach}$ state. If all the possible voxels are propagated and the voxels with $goal$ or $goal_{near}$ states are not reached, planning a path from the start to the goal for the robot with the set mobility is not possible.

D. Example

Here is a planned path example in a simple artificial environment. The experimental setup and path determined by our planner are shown in Fig. 5.

Fig. 6 and Fig. 7 show the state of layer 19 (heading direction θ^{yaw} = -18°) and layer 18 (heading direction θ^{yaw} = -36°), respectively. Table I defines the meanings of the colors in these figures.

A hill exists between the start and goal positions. It is impossible to climb up in a straight line because θ^{pitch}_{max} and θ^{roll}_{max} are smaller than the slope angle of the hill in this setup, as shown in Fig. 6. In contrast, there are no $obst$ state voxels and $obst_{near}$ state voxels in layer 18 because θ^{pitch} and θ^{roll} are smaller than θ^{pitch}_{max} and θ^{roll}_{max} in heading angle θ^{yaw}_{18} of this layer. Therefore, the planner plans a path to climb the hill diagonally.

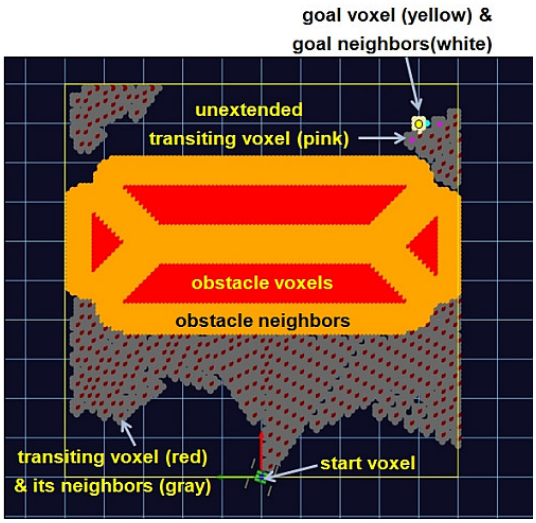

Fig. 6. Voxel states in layer 19 (θ_{yaw} = -18°)

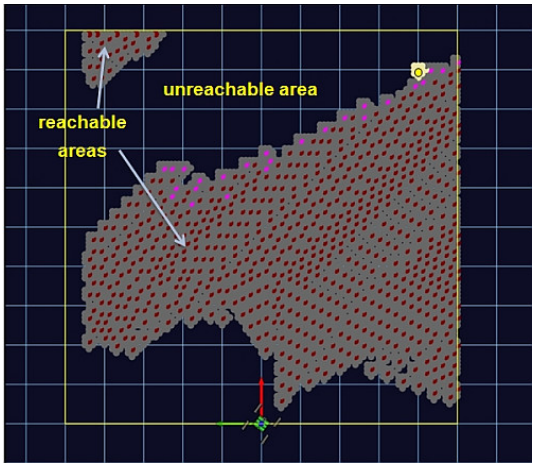

Fig. 7. Voxel states in layer 18 (θ_{yaw} = -36°)

III. Experimental Study

A. Objective

The objective of the experiments was to confirm the validity of the proposed method in a real volcano environment. In the experiments, the proposed method was first applied to the large volcano environment, and a path was planned from the set start position to the desired goal area. A small region in the vast field was then selected in order to confirm the validity of our method at a small scale. The two passing points of the obtained path in the vast field were selected as the local start and goal positions in the second experiment.

B. Target Environment

The target environment was a field near Mt. Nakadake, a mountain near an active volcano Mt. Shinmoedake in Kyushu, Japan (Fig. 8). Fig. 9 shows part of a DEM obtained

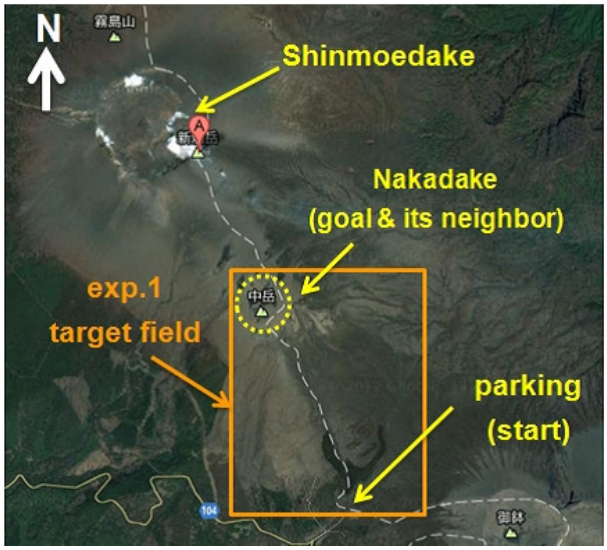

Fig. 8. Target environment around Nakadake/Shinmoedake

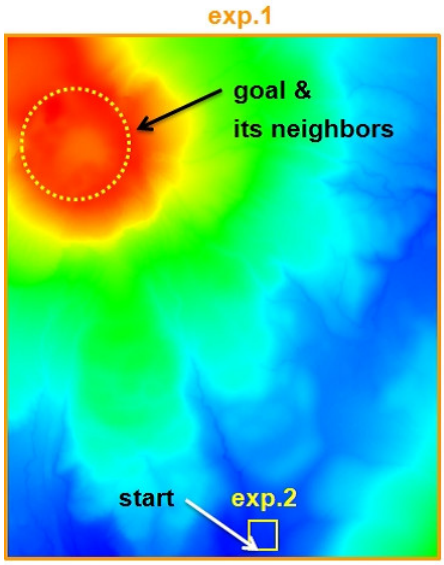

Fig. 9. Target field of experiment 1 and 2

by airplane LIDAR from the Kyushu Regional Bureau, Ministry of Land, Infrastructure, Transport and Tourism (MLIT). The resolution of the DEM is 1 m. In experiment 1, we set the global start position as the parking area near Nakadake and the global goal as the peak of the mountain. The target field size was 1.5 km in the east and west directions and 1.8 km in the north and south directions. As shown in Fig. 9, the target fields for experiments 1 and 2 were derived.

C. Implementation

We assumed the use of our mobile robot "El-Verde" (Fig. 1) in these experiments to fix the robot size. We implemented

TABLE II

PARAMETERS FOR PROPOSED METHOD

Code	Value in exp.1	Value in exp.2	Unit
l_x^{cell}, l_y^{cell}	16	1	m
l_x^{DEM}, l_y^{DEM}	1548, 1867	0.1, 0.1	m
x_m, y_m	97, 117	100, 100	-
θ_m^{yaw}	20	100, 100	-
θ_{max}^{roll}, θ_{max}^{pitch}	20 , 25	20, 25	°
R_{obst}	0.8	same as on the left	m
R_{goal}	12	3	cell
$R_{minturn}$	1	same as on the left	m
l_{seg}	3	same as on the left	cell
R_{reg}	2	same as on the left	cell
k_{dist}	0.002	0.03	-
k_{turn}	0.5	0.5	-
k_{pitch}	5	5	-
k_{roll}	5	5	-

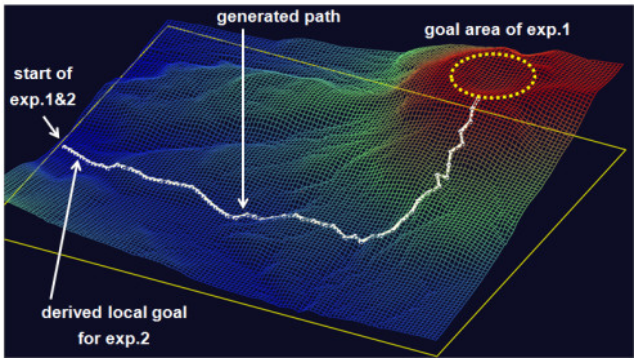

Fig. 10. Found path in experiment 1

the proposed method and selected the parameters listed in Table II for the experiments. The coefficients of the cost function were defined to maintain the maximum value of each term as almost 100 while considering their units. Please note the differences: the size of each cell in experiment 1 was 16 m, and that in experiment 2 was 1 m. The mobility parameters of the robot were empirically determined as the almost minimal-values that can obtain a path from the given start to the desired goal in experiment 1.

The proposed algorithm was implemented in Microsoft Visual C# language and executed on a computer having an Intel Core i7-3632QM processor as the CPU with a memory of 8 GB DDR3 RAM.

D. Results and Discussion

Fig. 10 shows the result of experiment 1. The proposed method generated a path from the start position to the desired area around the volcanic vent.

Based on the above results, we set the local goal position for experiment 2 as shown in Fig. 9 and Fig. 10. Fig. 11 shows the results of experiment 2. The derived path from the start position to the desired local goal was derived relatively straight.

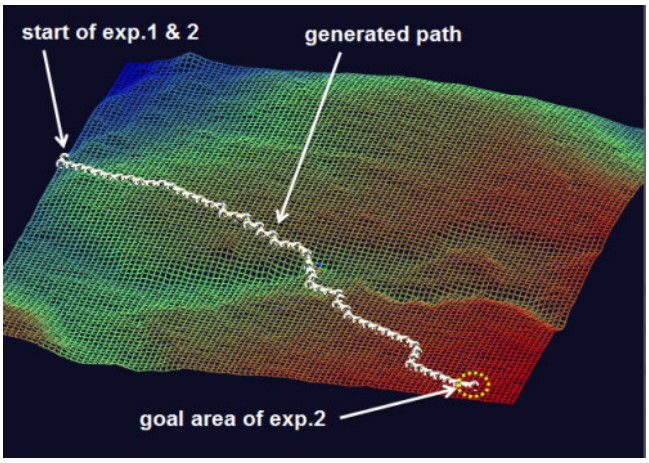

Fig. 11. Found path in experiment 2

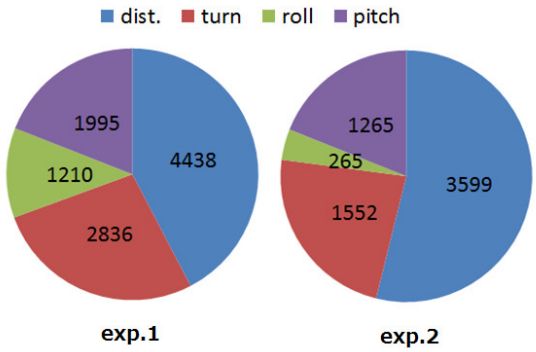

Fig. 12. Costs of path components in experiments 1 and 2

In experiment 1, a total of 29,109 voxels stored the transition-cost, and the time required for computation was 162 s. In experiment 2, a total of 34,976 voxels stored the transition-cost, and the time required for computation was 239 s. The time complexity depended on the terrain shape, mobility parameters of the robot such as θ_{max}^{roll} and θ_{max}^{pitch}, etc.

Fig. 12 shows the costs of the derived paths in experiments 1 and 2. Because the coefficients parameters in the cost function and the scales of the targeting fields differed in the two experiments, comparing the total costs was not important, but the costs of the components were of interest.

Because the environment smoothness differed, the composition ratio of each cost also differed in the two experiments. In experiment 1, the field contained precipitous mountains. Therefore, the planner planed a curving roundabout path to reduce θ^{pitch} and θ^{roll}. In contrast, the target field of experiment 2 was smooth. Therefore, the planned path was close to the minimum straight path from the start to the goal.

The results of the experimental study confirmed that the proposed method can derive a path even at the various scales/resolutions of a real volcano DEM obtained by airplane LIDAR.

IV. Conclusion

Ground mobile robot systems are desired to observe dangerous active volcanoes. To develop such robot systems, we presented a path planning technique for vast outdoor environments.

We focused on the use of a digital elevation map (DEM) obtained by airplane light detection and ranging (LIDAR) measured before volcanic eruptions. This airplane-based LIDAR can obtain vast terrain information quickly and efficiently. However, remote generation of sufficiently high-resolution DEM is difficult. Even if possible, path planning based on a high-resolution DEM of vast environments easily causes combinational explosion.

We proposed a path planning method that can be applied to any DEM resolution: path planning is seamlessly performed roughly at the vast scale and precisely at a local scale. The proposed method consists of three steps: DEM region and resolution selection, extended elevation map construction, and sparse transition-cost propagation. The proposed method was implemented in several experiments that were performed on a DEM around an active volcano in Japan.

Future research will involve performing a series of path planning experiments in real volcanic environments by combining DEMs obtained by airplane LIDAR and by a mobile robot's onboard LIDAR sensor.

Acknowledgment

This work is supported by the Kyushu Regional Bureau of the Ministry of Land, Infrastructure, Transport and Tourism (MLIT).

References

[1] T. Kaneko, T. Koyama, A. Yasuda, M. Takeo, T. Yanagisawa, K. Kajiwara, and Y. Honda, "Low-altitude remote sensing of volcanoes using an unmanned autonomous helicopter: an example of aeromagnetic observation at izu-oshima volcano, japan," *International Journal of Remote Sensing*, vol. 32, no. 5, pp. 1491–1504, 2011. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/01431160903559770

[2] A. L. Vedran Kordic and M. M. (Ed.), *A Robotic System for Volcano Exploration*. InTech, 2005.

[3] K. Nagatani, K. Hiroaki, K. Yoshida, T. Kenjiro, and K. Eiji, "Development of leg-track hybrid locomotion to traverse loose slopes and irregular terrain," in *Safety Security and Rescue Robotics (SSRR), 2010 IEEE International Workshop on*, 2010, pp. 1–6.

[4] T. Ohki, K. Sato, K. Nagatani, and K. Yoshida, "Development and evaluation of autonomous mobile manipulator for large scale outdoor environment," in *System Integration (SII), 2011 IEEE/SICE International Symposium on*, Dec. 2011, pp. 463–468.

[5] T. Kubota, Y. Kuroda, Y. Kunii, and T. Yoshimitsu, "Path planning for newly developed microrover," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 4, 2001, pp. 3710–3715.

[6] T. Ohki, K. Nagatani, and K. Yoshida, "Collision avoidance method for mobile robot considering motion and personal spaces of evacuees," in *Proc. of IEEE/RSJ International Conference on Intellegent Robots and Systems*, 2010.

[7] R. Jarvis and J. C. Byrne, "Robot navigation: Touching, seeing and knowing," in *Proc. of 1st Australian Conference on Artificial Intelligence*, 1986.